[redacted]

[redacted]

UIUCDCS-R-73-577

QUEUES AND NETWORK COMPUTERS

by

Linda Anne Da Rin Mills

May 18, 1973

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-73-577

QUEUES AND NETWORK COMPUTERS

by

Linda Anne Da Rin Mills

May 18, 1973

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

*Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign and supported in part by the National Science Foundation under grant No. GJ 28289.

ACKNOWLEDGMENT

The author would like to express her most sincere gratitude to her adviser, Professor E. K. Bowdon, Sr. His advice and criticisms provided untold support and encouragement for this paper.

My thanks are also due to my husband, Craig, for his many hours of discussion and proofreading.

Mrs. Gayanne Carpenter deserves a special thanks for her excellent work in typing this paper.

This research was supported in part by the National Science Foundation under Grant No. GJ 28289.

PREFACE

With the advent of the network computer has come an increasing movement towards developing the necessary analytical tools to properly evaluate and improve these networks. One such tool is system modeling to assist in performance evaluation of networks and computer centers whose usefullness and, consequently, communication flow is daily increasing. While much has been said of performance evaluation for particular centers, little has been done concerning the analysis of the entire network. Our purpose is to postulate a queueing theory model of a theoretical network and to use this model to predict such measures as the number in the system, the number in the queue, and system efficiency.

Under the assumptions of Poisson arrivals, exponential service times and a restricted queue length, the mathematical model is utilized to develop the steady state equations for the two-center and the n-center network. Then, a priority scheme is implemented and the steady state equations are rederived. With these two varying assumptions, conclusions are drawn as to their effect on system efficiency and the number in the queue.

Lastly, the relevance of proposing an analytical model is discussed. The mathematical model is compared to existing networks to verify its authenticity and accuracy. To the extent that such models aid our comprehension of the internal mechanisms and balances and thereby shape future networks, they serve a necessary and useful purpose.

TABLE OF CONTENTS

1. INTRODUCTION

Multiprocessor network systems have become a reality. With the reality comes the added burden of increasing speed and communication flow between centers. Networks offer little advantage over conventional processing techniques if centers operate with virtual independence of one another.

The expense of a network becomes feasible only if loads at one center can be merged with lighter loads at another center. Specialized services and skilled personnel, which are not economically viable for a single center, become viable when the larger population of potential users is taken into consideration. In addition, with much of the system saturation reduced, turnaround time can reach optimum levels. These are but a few of the advantages which can be gained by the emergence of the network as part of the computer technology.

Networks, such as ARPA or TUCC [8, 40] can be compared to the large business conglomerates. In both the essential need is to consolidate for reduction of overhead costs and amplification of profits. These two, business and networks, are more than similar, however, they are compatible. With the expanding market, the network offers an economic method of operating over a large data base in a geographically distributed region, thus reducing the amount of paperwork necessary in accounting and payroll, for example. However, not only are networks profitable for the large conglomerate, they are also advantageous to the smaller companies. This is their prime attraction.

With the emergence of networks to the computer technology, the next step is to provide analytic guidelines for the behavior of the networks, schemes for load leveling and algorithms to control phenomenon such as system saturation. Thus, networks must be analyzed in their entirety instead of being considered as a collection of disjoint centers. Without a doubt, this presents added problems and challenges. Load leveling involves assessing a situation which is never stationary and from that assessment deciding upon an appropriate strategy. Similarly, when a job enters the network, the center where this job will be processed must be determined so that such goals as economy and low turnaround time may be met. These goals may not coincide at any given point and any resolution of the conflict is complicated by the great number of decision points at any one moment.

However, a more important problem is the task of defining a general network structure that can be readily analyzed without too great a degree of simplification. A problem to which we now turn.

The simplest network consists of a single center. The flow through the center is determined by the arrival pattern of the commodity (customers or widgets) which utilize the center for a given period of time and the service pattern of the center. Clearly, if the customer demand exceeds the system capacity, then the center is unequipped to service this type of arrival pattern satisfactorily.

Consider the example of a single chair barber shop. If the customers arrive at a steady rate β, meaning in uniformly spaced time intervals, and the barber can cut hair at a uniform rate of μ, then $\beta \leq \mu$ means that the barber will be able to service all of his customers for that day. $\beta > \mu$ means that customers arrive faster than he can cut hair and a waiting line or queue will form. In this case, the queue will grow without bound (until closing time).

In constrast to the uniformly spaced arrivals, consider random or Poisson arrivals distributed around a mean β. This says that the exact arrival time of any one customer is unpredictable, but most will arrive within β of each other. Similarly, suppose the time it takes the barber to cut hair is randomly distributed around a mean of μ. Again, most hair cuts will take time μ to complete, but some may take a much longer or a much shorter time. Because both our barber and his customers are so unpredictable, it can be expected that a waiting line will form. This might happen, for instance, when two customers arrive a short time apart, before the barber has time to completely service the first arrival. At the same time, periods of idleness might exist, for instance, when two customers arrive far enough apart that the barber has finished with the first before the arrival of the second. However, given the means of β and μ, it can again be stated that when $\beta > \mu$, the waiting line will grow without bound. In other words, customers are on the average arriving faster than the barber can service them.

The network problems are even more interesting than these problems. One such network problem is the transportation problem, shown in Figure 1.1. A manufacturer wishes to ship a number of units of each widget from several factories to several warehouses. Each warehouse requires a certain number of units of each widget, while each factory can supply up to a certain amount. Only the routes indicated may be used for shipment and the problem is to minimize the transportation cost for shipment from factory to warehouse. Suppose, in addition, that warehouse #1 can never keep enough of A in stock and warehouse #2, enough of B. Then, the problem broadens to include maintaining a steady flow of these products to the specified warehouses. A solution of this type of problem is found in the literature [20].

Another example concerns the random arrivals or stochastic processes used earlier. For this example, suppose there exists a network of telephone lines whose topological structure is given in Figure 1.2. Each branch represents a two-way connection between cities, each with the capacity in number of calls per minute as indicated. Since the time and length of a call is unpredictable, this situation closely parallels our barber shop example. However, there are added complications in that the origin and destination of the call is not known beforehand, implying the existence of a store-and-forward mechanism. Again, as with computer networks one city's center may be overloaded and another idle. Each city has a center to process telephone calls, much like a computer center. Each call must queue up in a waiting line and wait transmission if the center is busy. If all the lines are busy, then the center is

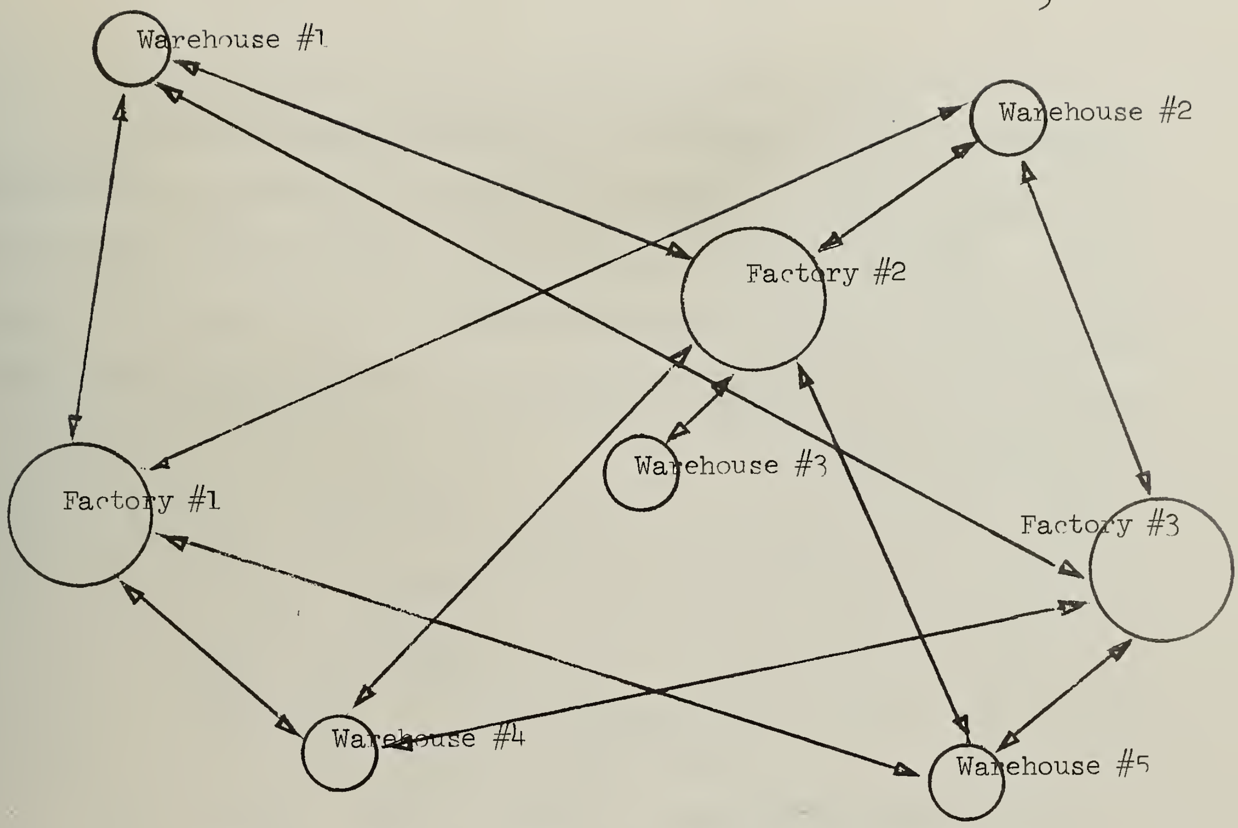


Figure 1.1. A Typical Network System with 3 Factories and 5 Warehouses

said to be operating at capacity and the <u>finite queue</u> is full. Suppose transmission of the call has an exponential service time Ω and suppose each call lasts an exponential time with mean μ. Then, this system can be viewed as being composed of a <u>dual server</u>, the transmission and communication servers. Since there is no mechanism to save incoming calls when the telephone is in use, a telephone has no queue on its communication server. We say that such calls during a busy period are <u>lost</u> to the system.

Furthermore, each time a person dials the phone he can view the destination as being determined by a probability matrix A which gives the probability that a call from him, center i, is placed at his neighbor's, center j. This is done with probability a_{ij}. Then if the number of calls he makes is randomly distributed around α, αa_{ij} is part of the total number of calls his neighbor is receiving. (Of course, they are receiving calls other than his.)

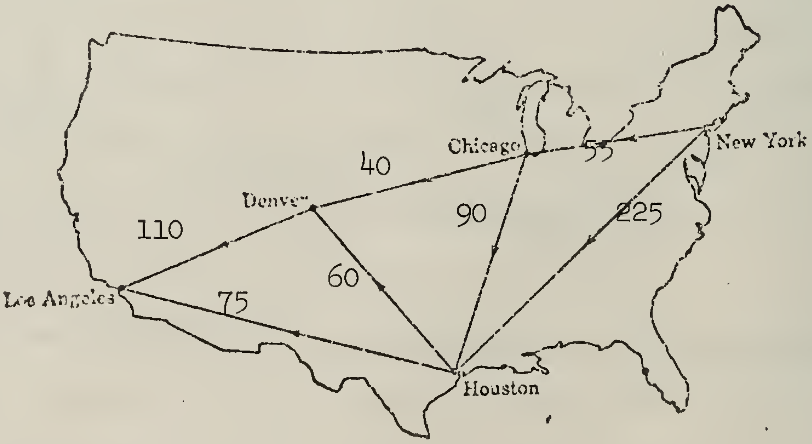


Figure 1.2. Distributed Telephone Network

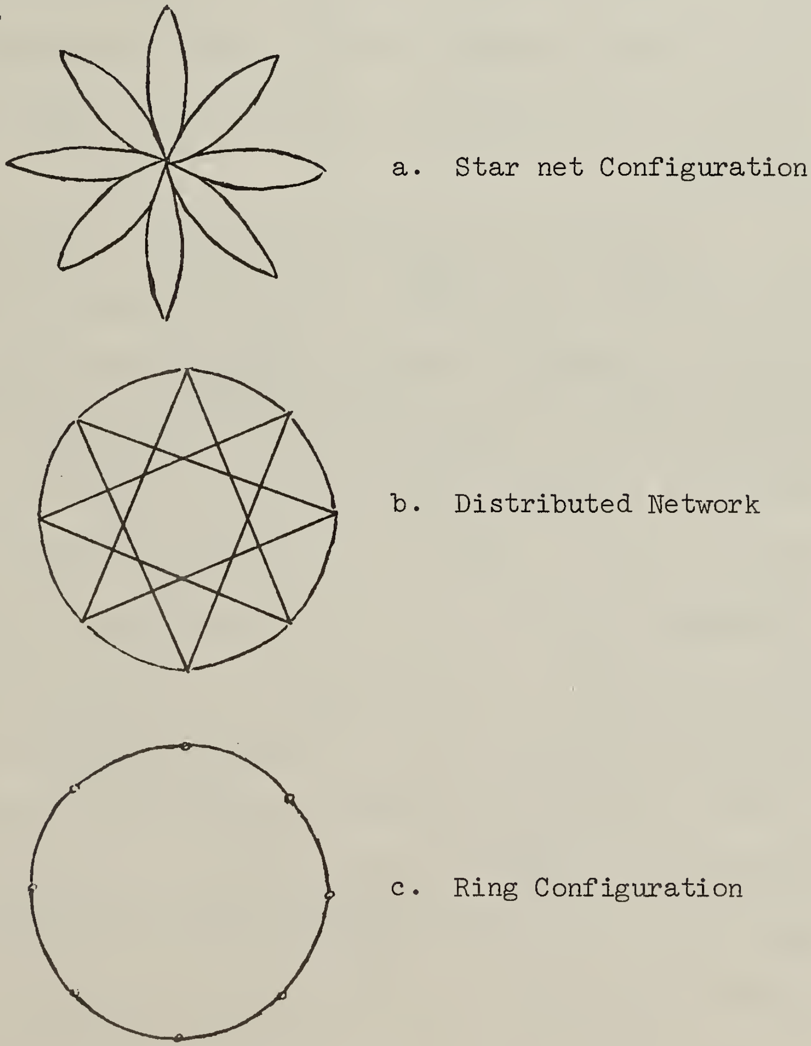
These examples separate into two groups: Networks and non-networks; random arrivals and uniform arrivals. It is to the problem of random arrivals in a network which we will speak. The solution to the barber shop problem with random arrivals is included in a broad class of queueing theory problems, of which the uniform arrivals represent a special case.

Specifically, we wish to focus our attention on the development of a general queueing theory model for a network. Once this has been accomplished, measures such as the number in the queue or system efficiency can be predicted. These measures are the real interest in any network, since if a network is to operate economically, it must not only function, but also show profits. The most obvious way to achieve profits is for the system manager to optimize system productivity. This is done successfully only if the manager has some idea of what is causing his bottleneck and why. Further, he needs to know how system changes will affect system utilization. For example, how increasing or decreasing job storage (disc or tape) will affect turnaround time.

Why networks? Hasn't this class of problems been considered previously? This may be true enough, but networks add a new dimension to these problems. A network computer has to decide when to transmit a job and when to process it on location. Often the user wants the quickest turnaround possible, how is this to be met in a network? Load leveling is now a necessity if centers are to cooperate within the network. Priorities come into play. Which job should have a high priority and which job a low one? Is the assignment a universal one or does it vary from center to center? How does a varying priority affect the computer manager's transmission decisions? The accounting and business management with the related data base complications opens up a new wealth of problems. Which center charges for the job processing--the initial or the end center? How is the integrity of the data base protected against unwelcome users? While these problems are mentioned for completeness, we wish to concern ourselves only with those problems in the realm of modeling and performance evaluation.

For convenience, we define a network center as a local and independent source of one or more processors. A network computer is defined to be an interconnected set of network centers. The transmitter for each center has an exponential service time and operates in the full duplex mode. Therefore, no messages are forced to wait for a free receiving transmitter. The exponential nature of the transmitter's service comes from the arbitrary nature of the transmitted job size. Clearly, these can be of any size and the length of the job would determine the transmitting time. The exponential assumption simply says that most jobs cluster around the mean, but there may exist very short or very long jobs. Furthermore, note that no rationale for why a job has to be transmitted is given at this time. We simply have said that part of those jobs submitted are transmitted without applying any rules or heuristics to decide what characteristics these jobs must have.

There are three network configurations which should be mentioned at this time. The first in Figure 1.3a is the star network, which consists of a center node which acts as the central message switching center. The outside nodes are characterized by one path leading in and out connecting it to the center node. TUCC is an illustration of this type of network topology. The distributed network, shown in Figure 1.3b, is distinguished by the number of alternate routes to any one node. It does not suffer from the vulnerability of the central switching node inherent to the star network. ARPA exemplifies this network philosophy. The third type (Figure 1.3c) is the ring structure, chosen for the Distributed Computer System. The major advantage gained by this topology is that the reduction of links still allows reliability and at the same time reduces costs. With this introduction, we now turn to the problem of model definition.



a. Star net Configuration

b. Distributed Network

c. Ring Configuration

Figure 1.3. Network Topology

2. NETWORK COMPUTER MODELING

2.1. Model Description

Suppose we are given a network computer of ℓ centers in the general form of Figure 2.1, and that the $i^{\underline{th}}$ center consists of c computers having an exponential service time distribution of μ_i. The input stream to the $i^{\underline{th}}$ center consists of two parts: those jobs destined for service completion at center i and those jobs destined for service completion at center j, $j \neq i$. Those jobs destined for service at center j arrive at center i from a Poisson distribution with an interarrival rate of α_i, and those jobs requesting service at center i arrive with an interarrival rate of β_i.

As determined by their destination, both types of input queue up in separate finite queues, called transmission and service queues, and wait for service. The finite queue assumptions are required since the storage capacity at each center is bounded. If the center is near saturation, clearly a portion of the jobs entering will be turned away for lack of space. (Since we are generally interested in the solution at steady state and not at the saturation point, the finite queue assumptions are not restrictive.) We say that the service queue at each center is bounded by length $N_i > 1$ and the transmission queue, by length $M_i > 1$, where N_i and M_i may differ from center to center. The transmitter for each center has an exponential service time of Ω_i, and operates in the full duplex mode. Therefore, no messages are forced to wait for a free receiving transmitter and no blocked state is assumed.

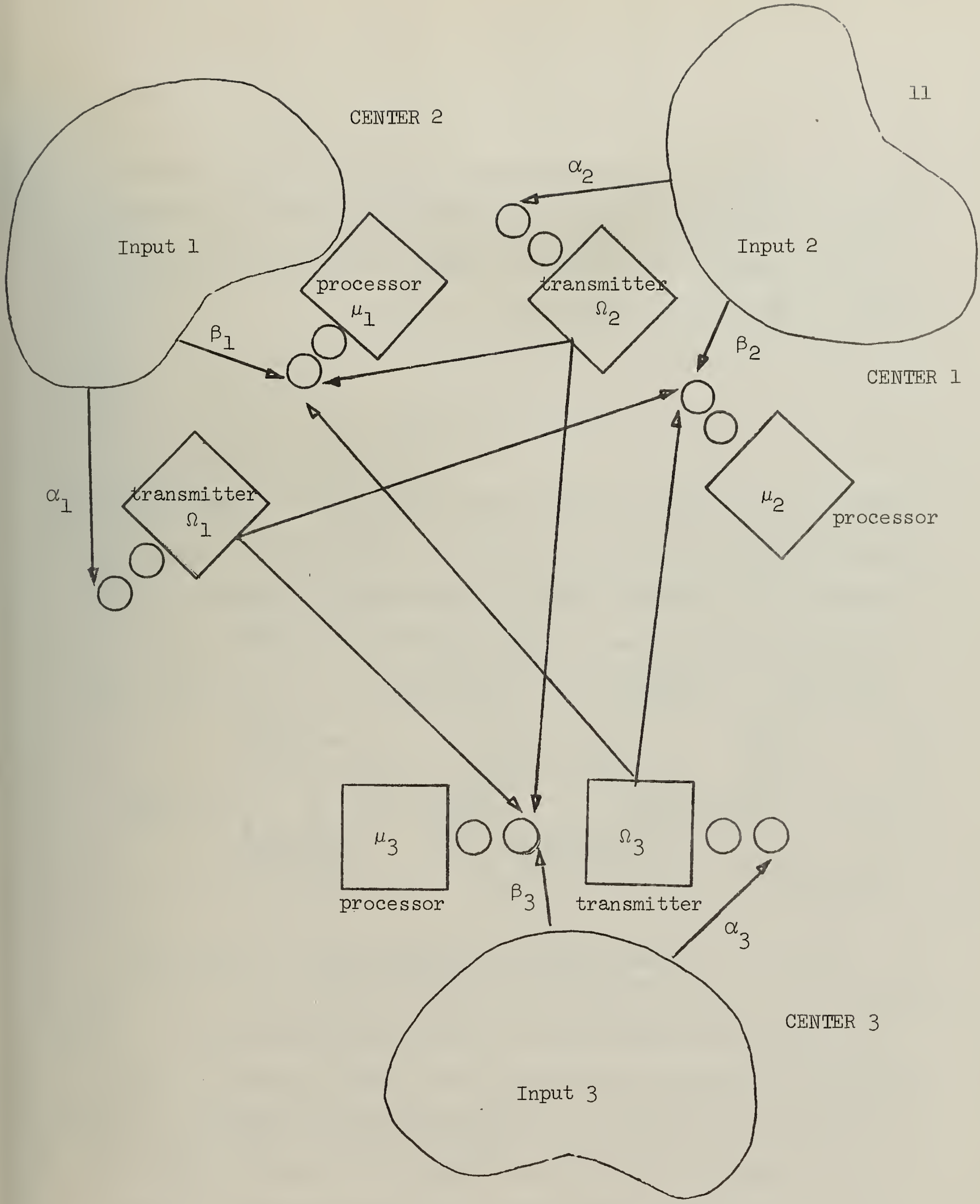


Figure 2.1. Three Center Network

Assume for the moment that no priorities between centers exist and that messages that are sent from center i to center j enter the finite queue for the server at center j. Then, the true arrival rate for center j, w_j, is given by

$$w_j = \beta_j + \sum_{i=1}^{\ell} \alpha_i a_{ij} \quad ,$$

where ℓ is the number of network centers and a_{ij} is the probability that the job is sent from center i to center j. Assume also that each of the ℓ centers has a compatible though not necessary equal capacity. Then, the following illustrates the queue structure for the i$^{\underline{th}}$ center.

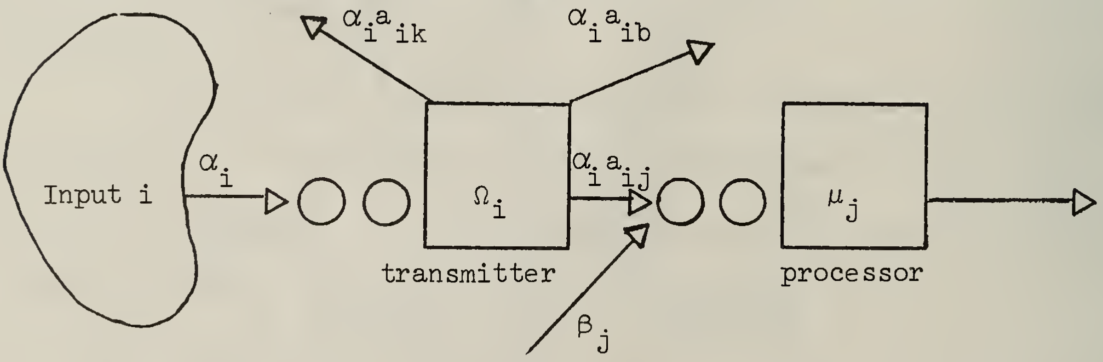


Figure 2.2. Queue at j$^{\underline{th}}$ center showing part of transmission input from other centers in net.

Our view of the network is much the same as that of an outside observer. We see jobs coming in at rate β_j and α_j to the i$^{\underline{th}}$ center. In addition to the stream β_j, our observer sees the stream $\alpha_i \; a_{ij}, i \neq j$, of jobs transmitted from the other centers in the network. These jobs queue up together with the stream β_j and together comprise the input for center j. Those jobs arriving at rate α_j are transmitted to center i, i $\neq$ j, at rate Ω_j, while those jobs remaining at center j complete service at rate, μ_j. Internal to this structure is the priority assignment, if any, and the load leveler for that particular center, etc.

Consider w_j.

$$w_j = \beta_j + \sum_{\substack{i=1 \\ i \neq j}}^{\ell} \alpha_i \, a_{ij}$$

$$= \beta_j + \alpha_i \, a_{ij} + \cdots + \alpha_{j-i} \, a_{j-i,j} + \alpha_{j+i} \, a_{j+i,j} + \cdots + \alpha_\ell a_{\ell j}.$$

(some of which may be zero)

This implies that each center has a probability matrix A of the form:

$$\begin{array}{c|ccccc|c}
i \backslash j & 1 & 2 & 3 & \cdots & \ell & \text{PROB. TOTAL} \\
\hline
1 & 0 & 1 & 2 & \cdots & 3 & 1 \\
2 & & 0 & & \cdots & & 1 \\
3 & & & 0 & \cdots & & 1 \\
\cdots & & & & & & \\
\ell & & & & & 0 & 1 \\
\hline
\end{array}$$

where a_{ij} is the probability that a job is sent from center i to center j.

Now for the case where $\ell = 2$, all transmitted units are from center 1 to center 2, and $a_{ij} = 1$, whenever i $\neq$ j. Then the probability matrix A simplifies to the following.

$$\begin{array}{c|cc|c}
 & 1 & 2 & \text{Total} \\
\hline
1 & 0 & 1 & 1 \\
2 & 1 & 0 & 1 \\
\hline
\end{array}$$

The arrival rate for center j, w_j, j = 1,2 becomes

$$w_j = \beta_j + \alpha_i \qquad i \neq j ,$$

or for compactness

$$w = \beta + \alpha$$

since for $\ell = 2$ the subscripts are unique for a given center.

In summary, we have assumed the following:

1. Input Population: We assume an infinite input population with a Poisson arrival rate of β_j for those jobs which arrive at center j and are serviced at center j. A Poisson arrival rate of α_j is assumed for those jobs arriving at center j and demanding service at center i, $i \neq j$. The arrival rate to the j^{th} processing center is

$$w_j = \beta_j + \sum_{\substack{i=1 \\ i \neq j}}^{\ell} \alpha_i \, a_{ij}$$

where ℓ is the number of centers.

2. Service Mechanism: Dual server consisting of a transmitter with an exponential service time of Ω_i at the i^{th} center and a processor with an exponential service time of μ_i at the i^{th} center.

3. Queue Discipline: At the i^{th} center, the transmitter has a queue bounded by length $M_i > 1$ and the processor has a queue of length $N_i > 1$. Entrance to the queue is first-come, first-served.

4. Routing Procedure: This is determined by the probability matrix A which gives the probability a_{ij} that a transmitted job is sent from center i to center j.

2.2. Temporal Behavior

Now that the network has been characterized, we turn to the derivation of the queueing model.

Consider the case where there are two centers in the network. Figure 2.3 illustrates the relationship between the transmitter and the processor. The exponential arrival rate for the processor at center 2 is β and μ is its service rate. As noted earlier, if a unit has been transmitted, it must come from center 1. The exponential arrival rate to the transmitter for center 1 is α and Ω is the service rate for the transmitter. (Note that for convenience of notation, we have deleted the subscripts from the processor for center 2 and the transmitter for center 1.) The dotted section of Figure 2.3 indicates the system under consideration.

The queue discipline is first-come , first-served and no preemption is permitted in either server. In the first server, the transmitter's queue length has been restricted to M; while in the second server, the queue is restricted to N. No blocking will occur in the transmitter at center 1 if center 2 has a queue of length N and a transmission is attempted. Units that enter when either queue is full are lost to the system.

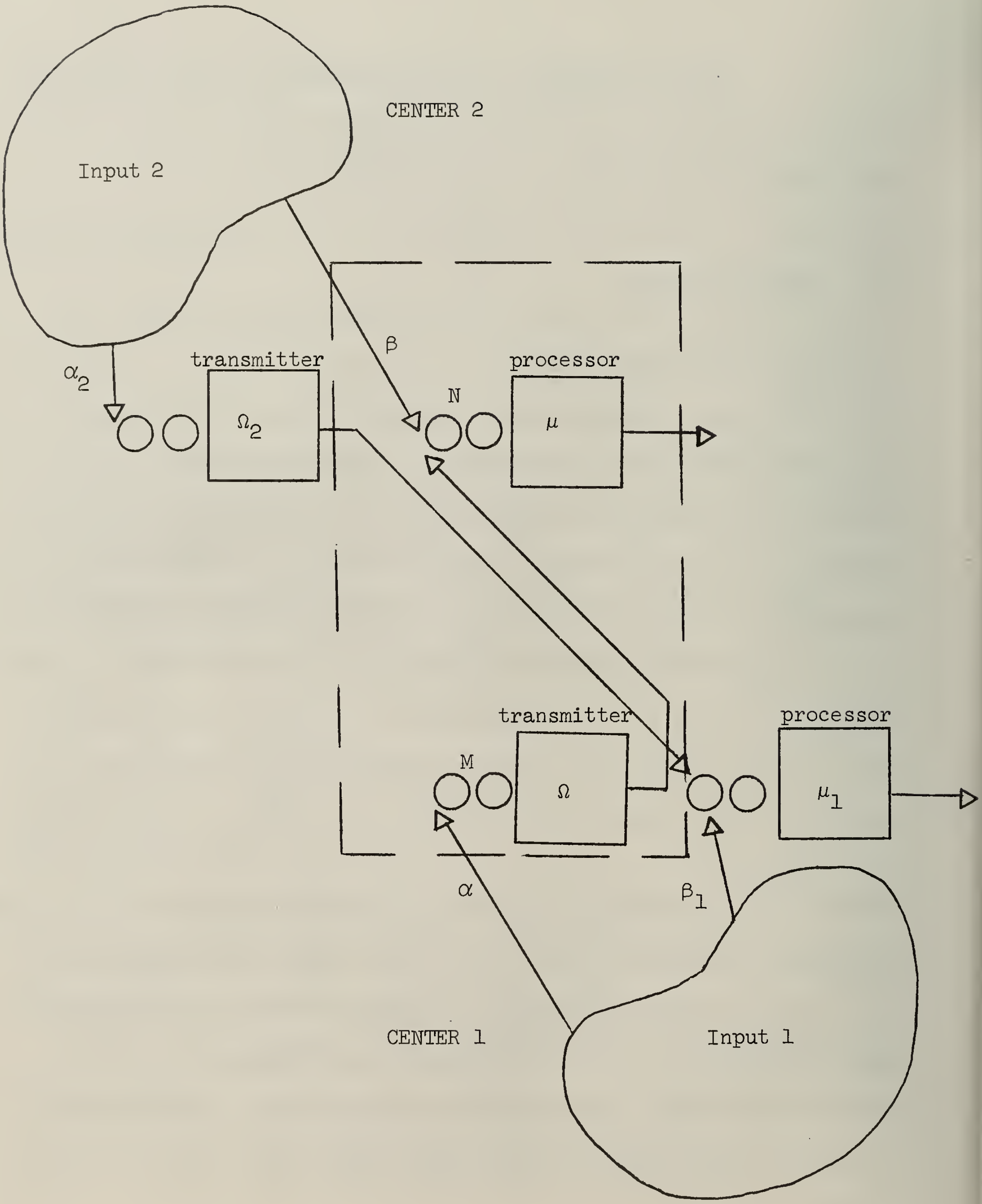


Figure 2.3. Two Center Network

One assumption should be mentioned at this point. The arrival rate to the transmitter is α. The output of the transmitter will also be α, if the queue (M) is infinite [7]. If this queue is not infinite, the input to center 2 will not necessarily be Poisson and in general, the arrival rate to center 2 from the transmitter will not be α. However, the arrival rate is assumed to be approximately α at steady state.

Let $P_{m,n}(t)$ denote the probability that at time t there are exactly m units in the first system and exactly n units in the second system at time t $(0 \leq m \leq M,\ 0 \leq n \leq N)$. We adopt the following notation:

1. w = effective arrival rate to center 2.

$$w = \alpha + \beta \quad .$$

2. ρ_T = traffic intensity at center 2.

$$\rho_T = w/\mu \quad .$$

3. ρ_1 = traffic intensity for the transmitter at center 1.

$$\rho_1 = \alpha/\Omega \quad .$$

4. ρ_2 = traffic intensity at center 2, exclusive of transmissions.

$$\rho_2 = \beta/\mu \quad .$$

5. L_s = average number in service.

6. η = system efficiency.

$$L_s/2 \quad .$$

7. E(m,n) = average number in system.

Note that the system is totally characterized by the probabilities $P_{m,n}$ at steady state.

The differential difference equations for the system are the following:

$$P_{0,0}'(t) = -(\alpha+\beta)\ P_{0,0}(t) + \mu\ P_{0,1}(t) \qquad (1)$$

$$P_{0,n}'(t) = -(\mu+\alpha+\beta)\ P_{0,n}(t) + \beta\ P_{0,n-1}(t) + \Omega\ P_{1,n-1}(t) + \mu\ P_{0,n+1}(t) \qquad (2)\ (n<N)$$

$$P_{m,0}'(t) = -(\alpha+\beta+\Omega)\ P_{m,0}(t) + \alpha\ P_{m-1,0}(t) + \mu\ P_{m,1}(t) \qquad (3)\ (m<M)$$

$$P_{m,n}'(t) = -(\alpha+\beta+\Omega+\mu)\ P_{m,n}(t) + \beta\ P_{m,n-1}(t) + \mu\ P_{m,n+1}(t) + \Omega\ P_{m+1,n-1}(t) + \alpha\ P_{m-1,n}(t) \qquad (4)\ (n<N,\ m<M)$$

$$P_{0,N}'(t) = -(\mu+\alpha)\ P_{0,N}(t) + \beta\ P_{0,N-1}(t) + \Omega\ P_{1,N-1}(t) + \Omega\ P_{1,N}(t) \qquad (5)\ (n=N)$$

$$P_{M,n}'(t) = -(\beta+\mu+\Omega)\ P_{M,n}(t) + \beta\ P_{M,n-1}(t) + \mu\ P_{M,n+1}(t) + \alpha\ P_{M-1,n}(t) \qquad (6)\ (m=M,\ n<N)$$

$$P_{m,N}'(t) = -(\alpha+\Omega+\mu)\ P_{m,N}(t) + \beta\ P_{m,N-1}(t) + \Omega\ P_{m+1,N-1}(t) + \alpha\ P_{m-1,N}(t) + \Omega\ P_{m+1,N}(t) \qquad (7)\ (m<M,\ n=N)$$

$$P_{M,0}'(t) = -(\beta+\Omega)\ P_{M,0}(t) + \mu\ P_{M,1}(t) + \alpha\ P_{M-1,0}(t) \qquad (8)\ (m=M)$$

$$P_{M,N}'(t) = -(\mu+\Omega)\ P_{M,N}(t) + \beta\ P_{M,N-1}(t) + \alpha\ P_{M-1,N}(t) \qquad (9)\ (m=M,\ n=N)$$

Equations (1) - (4) can be recognized as the differential difference equations for the dual infinite queue. Equations (1) - (4), (5), and (7) are the equations for the system (∞, N), where (∞, N) denotes an infinite queue for server 1 and a queue of length N for server 2. Equations (1) - (4), (6), and (8) are the differential difference equations for the (M, ∞) system. Equation (9) distinguishes the (M,N) system from the (M, ∞) system and the (∞, N) system.

Equation (7) (m in system 1 and n in system 2) results from the following argument. For $M>0$, an arrival at time $t + \Delta t$ will find m in system 1 and N in system 2 if and only if one of the following mutually exclusive and collectively exhaustive events occur:

1. If at time t there are m in system 1 and N in system 2, and in the interval Δt there are no arrivals in system 1 and no services in either server.

2. If at time t there are m in system 1 and N-1 in system 2, and in the interval Δt there were no arrivals to queue 1, one arrival to queue 2 and no services at either server.

3. If at time t there are m+1 in system 1 and N in system 2, and in the interval Δt there are no arrivals to either queue and no service to server 2, but server 1 completes a service, which is lost to the system since the queue for system 2 is full.

4. If at time t there are m+1 in system 1 and N-1 in system 2, and in the interval Δt there are no arrivals (α and β) to either queue from the outside and server 1 completes a service, while server 2 does not.

5. If at time t there are m-1 in system 1 and N in system 2, and in the interval Δt there is an arrival to queue 1 and none to queue 2, while neither server completes a service.

The following difference equation results from the above argument.

$$\begin{aligned} P_{m,N}(t + \Delta t) = {} & P_{m,N}(t)\ (1 - \alpha\ \Delta t)\ (1)\ (1 - \Omega\ \Delta t)\ (1 - \mu\ \Delta t) \\ & + P_{m,N-1}(t)\ (1 - \alpha - \Delta t)\ (1 - \mu\ \Delta t)\ (1 - \Omega\ \Delta t)\ \beta\ \Delta t \\ & + P_{m+1,N}(t)\ (1 - \alpha\ \Delta t)\ (1)\ (1 - \mu\ \Delta t)\ \Omega\ \Delta t \\ & + P_{m+1,N-1}(t)\ (1 - \alpha\ \Delta t)\ (1 - \beta\ \Delta t)\ (1 - \mu\ \Delta t)\ \Omega\ \Delta t \\ & + P_{m-1,N}(t)\ (1 - \Omega\ \Delta t)\ (1 - \mu\ \Delta t)\ (1)\ \alpha\ \Delta t \quad . \end{aligned}$$

Transposing the term $P_{m,n}(t)$, dividing by Δt, and taking the limit as $\Delta t \to 0$, we obtain equation (7).

To obtain the steady state probabilities, we set the left side of equations (1) - (9) to zero. Solving for the high order terms, and putting the results in a closed form yields the following solution:

$$P_{m,n} = \rho_1^{\ m} \rho_T^{\ n} P_{0,0} \qquad 0 \leq m < M,\ 0 \leq n \leq N \qquad \rho_1, \rho_T < 1 \qquad (10)$$

$$P_{M,n} = \rho_1^{\ M} \rho_2^{\ n} P_{0,0} \qquad m = M,\ 0 \leq n \leq N \qquad \rho_1, \rho_T < 1 \qquad (11)$$

where $P_{0,0}$ is determined as follows:

$$1 = \sum_{m=0}^{M} \sum_{n=0}^{N} P_{m,n} = \sum_{m=0}^{M-1} \sum_{n=0}^{N} P_{m,n} + \sum_{n=0}^{N} P_{M,n}$$

$$P_{0,0}^{\ -1} = \frac{((1-\rho_2)(1-\rho_T^{\ N+1})(1-\rho_1^{\ M}) + \rho_1^{\ M}(1-\rho_2^{\ N+1})(1-\rho_T)(1-\rho_1))}{(1-\rho_T)(1-\rho_1)(1-\rho_2)} \qquad (12)$$

where $\rho_1,\ \rho_2,\ \rho_T < 1$.

2.3. System Descriptors

The average number in service is:

$$\begin{aligned}
L_s &= 0\cdot P_{0,0} + \sum_{n=1}^{N} 1\cdot P_{0,n} + \sum_{m=1}^{M} 1\cdot P_{m,0} + \sum_{m=1}^{M}\sum_{n=1}^{N} 2\cdot P_{m,n} \\
&= P_{0,0}\Big[(1-\rho_2)(\rho_T-\rho_T^{N+1}) + (1-\rho_2)(\rho_1-\rho_1^{M+1}) + 2\rho_1^{M}\rho_2(1-\rho_1) \\
&\quad - 2\rho_1^{M}\rho_T^{N+1}\rho_2 + \rho_T\rho_1^{M+1}(1+\rho_2) - 2\rho_1^{M}\rho_2^{N+1}(1-\rho_1)(1-\rho_T) \\
&\quad - \rho_T^{N+1}\rho_1(1-\rho_2) - 2\rho_1^{M}\rho_T(1-\rho_T^{N})\Big] \Big/ \Big[(1-\rho_1)(1-\rho_2)(1-\rho_T)\Big]
\end{aligned} \tag{13}$$

where ρ_1, ρ_2, and $\rho_T < 1$.

The average number in System 1:

$$\begin{aligned}
E(m) &= \sum_{m=1}^{M} mP_m = \sum_{m=1}^{M} m \sum_{n=0}^{N} P_{m,n} \\
&= \left[\frac{(1-\rho_T^{N+1})[\rho_1(1-\rho_1^{M}) - M\rho_1^{M}(1-\rho_1)]}{(1-\rho_T)(1-\rho_1)^2} + \frac{M\rho_1^{M}(1-\rho_2^{N+1})}{(1-\rho_2)}\right] P_{0,0}
\end{aligned} \tag{14}$$

where ρ_1, $\rho_T < 1$.

The average number in System 2 is:

$$
\begin{aligned}
E(n) &= \sum_{n=1}^{N} nP_n = \sum_{n=1}^{N} n \sum_{m=0}^{M} P_{m,n} \\
&= \left[\frac{(1-\rho_1{}^{M})[\rho_T(1-\rho_T{}^{N+1}) - (N+1)(1-\rho_T)\rho_T{}^{N+1}]}{(1-\rho_1)(1-\rho_T)^2} \right. \\
&\quad \left. + \frac{\rho_1{}^{M}[\rho_2(1-\rho_2{}^{N+1}) - (N+1)(1-\rho_2)\rho_2{}^{N+1}]}{(1-\rho_2)^2} \right] P_{0,0}
\end{aligned}
\tag{15}
$$

where $\rho_1,\ \rho_2,\ \rho_T < 1.$

The average number in the system is given by the following sum:

1. Expected number in the transmitter's system, $E(m)$,

2. Expected number in the processor's system, $E(n)$.

Summing, we obtain

$$
\begin{aligned}
E(m,n) &= E(m) + E(n) = \sum_{m=1}^{M} m \sum_{n=0}^{N} P_{m,n} + \sum_{n=1}^{N} n \sum_{m=0}^{M} P_{m,n} \\
&= \left[\frac{(1-\rho_T{}^{N+1})[\rho_1(1-\rho_1{}^{M}) - M\rho_1{}^{M}(1-\rho_1)]}{(1-\rho_T)(1-\rho_1)^2} \right. \\
&\quad + \frac{M\rho_1{}^{M}(1-\rho_2{}^{N+1})}{(1-\rho_2)} + \frac{(1-\rho_1{}^{M})[\rho_T(1-\rho_T{}^{N+1})-(N+1)(1-\rho_T)\rho_T{}^{N+1}]}{(1-\rho_1)(1-\rho_T)^2} \\
&\quad \left. + \frac{\rho_1{}^{M}[\rho_2(1-\rho_2{}^{N+1}) - (N+1)(1-\rho_2)\rho_2{}^{N+1}]}{(1-\rho_2)^2} \right] P_{0,0}
\end{aligned}
\tag{16}
$$

where $\rho_1,\ \rho_T < 1.$

The average number in the transmitter's queue is:

$$L_{q1} = \sum_{m=1}^{M} (m-1) \sum_{n=0}^{N} P_{m,n}$$

$$= \left[\frac{(1-\rho_T^{N+1})[\rho_1^{2}(1-\rho_1^{M-1})-M\rho_1^{M}(1-\rho_1)+\rho_1^{M}(1-\rho_1)]}{(1-\rho_T)(1-\rho_1)^{2}} + \frac{(M-1)\rho_1^{M}(1-\rho_2^{N+1})}{(1-\rho_2)} \right] P_{0,0} \tag{17}$$

where ρ_1, $\rho_T < 1$.

The average number in the processor's queue is:

$$L_{q2} = \sum_{n=1}^{N} (n-1) \sum_{m=0}^{M} P_{m,n}$$

$$= \left[\frac{(1-\rho_1^{M})[\rho_T^{N+1}(1-\rho_T)-(N+1)\rho_T^{N+1}(1-\rho_T)+\rho_T^{2}(1-\rho_T^{N})]}{(1-\rho_1)(1-\rho_T)^{2}} \right.$$

$$\left. + \frac{\rho_1^{M}[\rho_2^{2}(1-\rho_2^{N})-(N+1)\rho_2^{N+1}(1-\rho_2)+\rho_2^{N+1}(1-\rho_2)]}{(1-\rho_2)^{2}} \right] P_{0,0} \tag{18}$$

where ρ_1, $\rho_T < 1$.

The total number in the queue L_q can be expressed as the sum

$$L_q = L_{q1} + L_{q2}.$$

For System 1:

$$P_1(>0) = \sum_{m=1}^{M} \sum_{n=0}^{N} P_{m,n} \quad .$$

This is just the probability that there are one or more jobs in System 1.

$$p_1(>0) = \left[\frac{(1-\rho_1^{M-1})\rho_1(1-\rho_T^{N+1})}{(1-\rho_1)(1-\rho_T)} + \frac{\rho_1^{M}(1-\rho_2^{N+1})}{(1-\rho_2)} \right] P_{0,0} \qquad (19)$$

The average waiting times are derived from the following argument. A job entering System 1 will find η jobs ahead of it. After these η have been served, it will be first in the queue and must then wait for one more service before it can go into service. So,

$$\eta = L_{q1}/p_1(>0) \text{ and } W_1 = (\eta+1)/\Omega = (L_{q1}/p_1(>0) + 1)/\Omega \quad .$$

This is the waiting time given that the job waits (i.e. $W_1 | p_1(>0)$).
Therefore,

$$W_1 = [L_{q1} + p_1(>0)]/\Omega \quad .$$

$$= \left[\frac{(1-\rho_T^{N+1})[\rho_1^{M}(1-\rho_1)-M(1-\rho_1)\rho_1^{M}+\rho_1(1-\rho_1^{M-1})]}{(1-\rho_T)(1-\rho_1)^2} + \frac{M\rho_1^{M}(1-\rho_2^{N+1})}{(1-\rho_2)} \right] P_{0,0} \, /\Omega \qquad (20)$$

where $\rho_1, \rho_T < 1$.

For System 2,

$$p_2(>0) = \sum_{n=1}^{N} \sum_{m=0}^{M} P_{m,n} \quad .$$

This is the probability that there are one or more jobs in System 2.

$$p_2(>0) = \left[\frac{(1-\rho_1{}^M)\rho_T(1-\rho_T{}^N)}{(1-\rho_1)(1-\rho_T)} + \frac{\rho_1{}^M \rho_2(1-\rho_2{}^N)}{(1-\rho_2)} \right] P_{0,0} \tag{21}$$

Similarly, with ρ_1, $\rho_T < 1$

$$\begin{aligned} W_2 = {} & [L_{q2} + p_2(>0)]/\mu \\ = {} & \left[\frac{(1-\rho_1{}^M)[\rho_T(1-\rho_T{}^N)-(N+1)\rho_T{}^{N+1}(1-\rho_T)+\rho_T{}^{N+1}(1-\rho_T)]}{(1-\rho_1)(1-\rho_T)^2} \right. \\ & \left. + \frac{\rho_1{}^M[\rho_2(1-\rho_2{}^N)-(N+1)\rho_2{}^{N+1}(1-\rho_2)+\rho_2{}^{N+1}(1-\rho_2)]}{(1-\rho_2)^2} \right] P_{0,0}/\mu \end{aligned} \tag{22}$$

In summary, the total waiting time of a job for the system is dependent on where it enters the system. If it joins the system as part of the stream β, then the expected waiting time will be $W = W_2$, but if the job joins as part of the stream α, then the expected waiting time will be $W = W_1 + W_2$. As can be seen, the total waiting time for any job is dependent on the number of transmissions it experiences before it is served.

2.4. Limiting Cases

Case 1 (M, ∞):

Letting $N \to \infty$ in equations (10) and (11) does not affect the solution set for this system, since (10) and (11) are independent of N. Equation (9) distinguishes the (M,N) system from the (M,∞) system, where the set of equations for the system (M, ∞) is a subset of those for the system (M,N). This further supports the conclusion of equivalent solution sets.

Equation (12) becomes:

$$P_{0,0}^{-1} = \frac{(1-\rho_2)(1-\rho_1^M) + \rho_1^M(1-\rho_1)(1-\rho_T)}{(1-\rho_T)(1-\rho_1)(1-\rho_2)}$$

where ρ_1, ρ_2, and $\rho_T < 1$.

A summary of results for the system (M, ∞) can be found in Appendix A.

Case 2(∞, N):

In the limit as M→∞ in equation (10) and (11), we obtain the solution for the system (∞, N). Since

$$\lim_{M\to\infty} \rho_1^M \rho_2^n P_{0,0} = 0 \quad ,$$

the solution set consists of equation (10) alone, where

$$P_{0,0}^{-1} = (1-\rho_T^{N+1})/(1-\rho_1)(1-\rho_T)$$

with $\rho_1, \rho_T < 1$. A summary of the results for the system descriptors can be found in Appendix B.

Case 3(∞, ∞):

Letting M→∞ and N→∞ in equations (10) and (11) produces a solution set of equation (10) for the system (∞, ∞) where

$$P_{0,0} = (1-\rho_T)(1-\rho_1)$$

with $\rho_1, \rho_T < 1$. This is the well-known result of Morse [28] and Saaty [35]. A summary of the results for the system descriptors can be found in Appendix C.

3. GENERAL MODEL

The two center case suffers from one failing and that is the small number of centers over-simplifies the probability matrix A. As a result, none of the stream splitting from the output stream of the transmitter occurs. We will develop next a three center model and from that generalize to the n-center model. For ease of computation and notation, the probability matrix A will not be immediately implemented. Specifically, we assume, for the present:

1. All of center 1's transmitted jobs are sent to center 2, (i.e. $\alpha_{12} = 1$), and

2. All of center 3's transmitted jobs are sent to center 2, (i.e. $\alpha_{13} = 1$).

Figure 2.1 and 3.1 illustrates the three center case with matrix A implemented and with the assumptions above, respectively. The dotted section in Figure 3.1 indicates the system presently under consideration.

The exponential arrival rate for the processor at center 2 is μ_2 and the transmission rate for centers 1 and 3 is Ω_1 and Ω_3. The arrival rates for the transmitters are Poisson around a mean of α_1, i = 1,3, and the arrival rate for the processor at center 2 is β_2. The queues for the transmitters at centers 1 and 3 are restricted to K>1 and M>1,

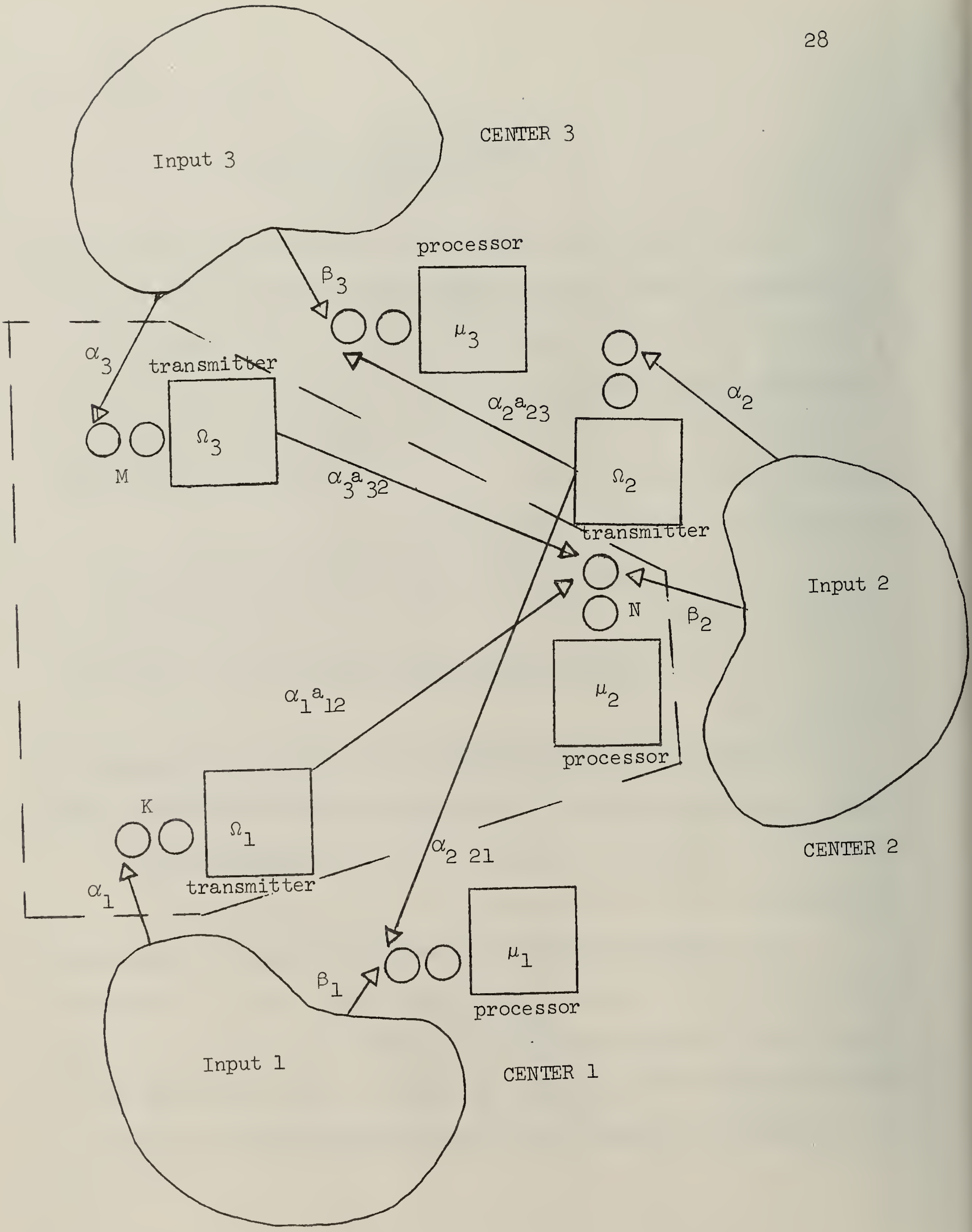


Figure 3.1. Three Center Model

respectively, while the queue for the processor at center 2 is restricted to $N>1$. Units that enter when the queues are full are lost to the system, as previously.

Let $P_{k,m,n}(t)$ denote the probability that there are exactly k units in System 1 at time t, exactly m units in System 3 at time t, and exactly n units in System 2 at time t ($0 \le k \le K$, $0 \le m \le M$, $0 \le n \le N$). We adopt the following notation:

1. w = true arrival rate to center 2

$$w = \alpha_1 + \alpha_3 + \beta_2.$$

2. ρ_T= total traffic intensity at center 2

$$\rho_T = w/\mu_2.$$

3. ρ_1= traffic intensity for transmitter at center 1

$$\rho_1 = \alpha_1/\Omega_1.$$

4. ρ_2= traffic intensity for the processor at center 2, exclusive of transmissions

$$\rho_2 = \beta_2/\mu_2.$$

5. ρ_3= traffic intensity for transmitter at center 3

$$\rho_3 = \alpha_3/\Omega_3.$$

Note that the probabilities $P_{k,m,n}$ totally characterize the system at steady state. Continuing as earlier, the differential difference equations for the system (K,M,N) are:

$$P_{0,0,0}'(t) = -(\alpha_1 + \alpha_3 + \beta_2)\, P_{0,0,0}(t) + \mu_2 P_{0,0,1}(t) \qquad (23)$$

$$\begin{aligned} P_{0,0,n}'(t) = {} & -(\alpha_1 + \alpha_3 + \beta_2 + \mu_2)\, P_{0,0,n}(t) + \beta_2 P_{0,0,n-1}(t) \\ & + \mu_2 P_{0,0,n+1}(t) + \Omega_3 P_{0,1,n-1}(t) \\ & + \Omega_1 P_{1,0,n-1}(t) \qquad (n < N) \qquad (24) \end{aligned}$$

$$\begin{aligned} P_{0,m,0}'(t) = {} & -(\alpha_1 + \alpha_3 + \beta_2 + \Omega_1)\, P_{0,m,0}(t) \\ & + \alpha_3 P_{0,m-1,0}(t) + \mu_2 P_{0,m,1}(t) \qquad (m < M) \qquad (25) \end{aligned}$$

$$\begin{aligned} P_{k,0,0}'(t) = {} & -(\alpha_1 + \alpha_3 + \beta_2 + \Omega_3)\, P_{k,0,0}(t) \\ & + \alpha_1 P_{k-1,0,0}(t) + \mu_2 P_{k,0,1}(t) \qquad (k < K) \qquad (26) \end{aligned}$$

$$\begin{aligned} P_{k,m,n}'(t) = {} & -(\alpha_1 + \alpha_3 + \beta_2 + \Omega_1 + \Omega_3 + \mu_2)\, P_{k,m,n}(t) \\ & + \beta_2 P_{k,m,n-1}(t) + \mu_2 P_{k,m,n+1}(t) \\ & + \Omega_3 P_{k,m+1,n-1}(t) + \Omega_1 P_{k+1,m,n-1}(t) \qquad \begin{pmatrix} m < M \\ n < N \\ k < K \end{pmatrix} \qquad (27) \\ & + \alpha_3 P_{k,m-1,n}(t) + \alpha_1 P_{k-1,m,n}(t) \end{aligned}$$

$$\begin{aligned} P_{0,0,N}'(t) = {} & -(\alpha_1 + \alpha_3 + \mu_2)\, P_{0,0,N}(t) + \beta_2 P_{0,0,N-1}(t) \\ & + \Omega_3 P_{0,1,N-1}(t) + \Omega_1 P_{1,0,N-1}(t) \\ & + \Omega_3 P_{0,1,N}(t) + \Omega_1 P_{1,0,N}(t) \qquad (n = N) \qquad (28) \end{aligned}$$

$$\begin{aligned} P_{0,M,0}'(t) = {} & -(\alpha_3 + \beta_2 \ \Omega_1) \ P_{0,M,0}(t) + \alpha_3 P_{0,M-1,0}(t) \\ & + \mu_2 P_{0,M,1}(t) \qquad (m = M) \end{aligned} \tag{29}$$

$$\begin{aligned} P_{K,0,0}'(t) = {} & -(\alpha_1 + \beta_2 + \Omega_3) \ P_{K,0,0}(t) + \alpha_1 P_{K-1,0,0}(t) \\ & + \mu_2 P_{K,0,1}(t) \qquad (k = K) \end{aligned} \tag{30}$$

$$\begin{aligned} P_{k,M,n}'(t) = {} & -(\alpha_3 + \beta_2 + \mu_2 + \Omega_1 + \Omega_3) \ P_{k,M,n}(t) \\ & + \beta_2 P_{K,M,n-1}(t) + \mu_2 P_{k,M,n+1}(t) \\ & + \alpha_3 P_{k,M-1,n}(t) + \alpha_1 P_{k-1,M,n}(t) \qquad \begin{pmatrix} k < K \\ m = M \\ n < N \end{pmatrix} \\ & + \Omega_1 P_{k+1,M,n-1}(t) \end{aligned} \tag{31}$$

$$\begin{aligned} P_{K,m,n}'(t) = {} & -(\alpha_1 + \beta_2 + \mu_2 + \Omega_1 + \Omega_3) \ P_{K,m,n}(t) \\ & + \beta_2 P_{K,m,n-1}(t) + \mu_2 P_{K,m,n+1}(t) \\ & + \alpha_1 P_{K-1,m,n}(t) + \alpha_3 P_{K,m-1,n}(t) \qquad \begin{pmatrix} k = K \\ m < M \\ n < N \end{pmatrix} \\ & + \Omega_3 P_{K,m+1,n-1}(t) \end{aligned} \tag{32}$$

$$\begin{aligned} P_{k,m,N}'(t) = {} & -(\alpha_1 + \alpha_3 + \mu_2 + \Omega_1 + \Omega_3) \ P_{k,m,N}(t) \\ & + \beta_2 P_{k,m,N-1}(t) + \Omega_3 P_{k,m+1,N-1}(t) \\ & + \Omega_1 P_{k+1,m,N-1}(t) + \alpha_3 P_{k,m-1,N}(t) \qquad \begin{pmatrix} k < K \\ m < M \\ n = N \end{pmatrix} \\ & + \alpha_1 P_{k-1,m,N}(t) + \Omega_3 P_{k,m+1,N}(t) \\ & + \Omega_1 P_{k+1,m,N}(t) \end{aligned} \tag{33}$$

$$\begin{aligned} P_{K,M,N}'(t) = {} & -(\mu_2 + \Omega_1 + \Omega_3)\ P_{K,M,N}(t) + \beta_2 P_{K,M,N-1}(t) \\ & + \alpha_3 P_{K,M-1,N}(t) + \alpha_1 P_{K-1,M,N}(t) \end{aligned} \qquad \begin{pmatrix} k = K \\ m = M \\ n = N \end{pmatrix} \tag{34}$$

$$\begin{aligned} P_{K,M,n}'(t) = {} & -(\beta_2 + \Omega_1 + \Omega_3 + \mu_2)\ P_{K,M,n}(t) + \alpha_3 P_{K,M-1,n}(t) \\ & + \alpha_1 P_{K-1,M,n}(t) + \mu_2 P_{K,M,n+1}(t) \\ & + \beta_2 P_{K,M,n-1}(t) \end{aligned} \qquad \begin{pmatrix} k = K \\ m = M \\ n < N \end{pmatrix} \tag{35}$$

$$\begin{aligned} P_{K,m,N}'(t) = {} & -(\alpha_1 + \Omega_1 + \Omega_3 + \mu_2)\ P_{K,m,N}(t) + \alpha_3 P_{K,m-1,N}(t) \\ & + \alpha_1 P_{K-1,m,N}(t) + \beta_2 P_{K,m,N-1}(t) \\ & + \Omega_3 P_{K,m+1,N-1}(t) + \Omega_3 P_{K,m+1,N}(t) \end{aligned} \qquad \begin{pmatrix} k = K \\ m < M \\ n = N \end{pmatrix} \tag{36}$$

$$\begin{aligned} P_{k,M,N}'(t) = {} & -(\alpha_3 + \Omega_1 + \Omega_3 + \mu_2)\ P_{k,M,N}(t) + \beta_2 P_{k,M,N-1}(t) \\ & + \alpha_3 P_{k,M-1,N}(t) + \Omega_1 P_{k+1,M,N-1}(t) \\ & + \alpha_1 P_{k-1,M,N}(t) + \Omega_1 P_{k+1,M,N}(t) \end{aligned} \qquad \begin{pmatrix} k < K \\ m = M \\ n = N \end{pmatrix} \tag{37}$$

Equations (12) - (26) are derived as equation (7) was in the two center development. If we set the left side of (12) - (26) to zero, then the steady state solution to this set of equations is as follows:

$$P_{k,m,n} = \rho_1{}^k \rho_3{}^m \rho_T{}^n P_{0,0} \qquad 0 \le k < K \;,\; 0 \le m < M \;,\; 0 \le n \le N \tag{38}$$

$$P_{k,M,n} = \rho_1{}^k \rho_3{}^M \rho_i{}^n P_{0,0} \qquad 0 \le k < K \;,\; m=M \;,\; 0 \le n \le N \qquad \rho_i = (\beta_2 + \alpha_1)/\mu_2 \tag{39}$$

$$P_{K,m,n} = \rho_1{}^K \rho_3{}^m \rho_j{}^n P_{0,0} \qquad k=K \;,\; 0 \le m < M \;,\; 0 \le n \le N \qquad \rho_j = (\beta_2 + \alpha_3)/\mu_2 \tag{40}$$

$$P_{K,M,n} = \rho_1{}^K \rho_3{}^M \rho_2{}^n P_{0,0} \qquad k=K \;,\; m=M \;,\; 0 \le n \le N \tag{41}$$

where $P_{0,0,0}{}^{-1}$ is as follows:

$$\begin{aligned} P_{0,0,0}{}^{-1} = \Big[& (1-\rho_T{}^{N+1})(1-\rho_1{}^M)(1-\rho_3{}^K)(1-\rho_2)(1-\rho_j)(1-\rho_i) + (1-\rho_2{}^{N+1})(1-\rho_T)(1-\rho_1) \\ & (1-\rho_3)(1-\rho_i)(1-\rho_j)\rho_3{}^K \rho_1{}^M + \rho_1{}^M(1-\rho_i{}^{N+1})(1-\rho_3{}^K)(1-\rho_j)(1-\rho_2)(1-\rho_T) \\ & (1-\rho_1) + \rho_3{}^K(1-\rho_j{}^{N+1})(1-\rho_1{}^M)(1-\rho_3)(1-\rho_T)(1-\rho_2)(1-\rho_i) \Big] \Big/ \\ & \Big[(1-\rho_T)(1-\rho_1)(1-\rho_2)(1-\rho_3)(1-\rho_i)(1-\rho_j) \Big] \end{aligned}$$

with ρ_1, ρ_2, ρ_3, ρ_i, ρ_j, and $\rho_T < 1$.

As a limiting case, consider the system (∞, ∞, ∞), then

$$P_{0,0,0}{}^{-1} = 1/(1-\rho_T)(1-\rho_1)(1-\rho_3)$$

as expected, where ρ_1, ρ_3 and $\rho_T < 1$. Limiting cases for the other possible systems can be developed as earlier. These derivations are left as an exercise for the reader.

The interesting point to be noted here is that as extra input streams are added to the model, the complexity of the model is enormously increased both in the number of equations which must be considered and in the expression for $P_{0,0,0}^{-1}$ itself. The number of equations to be considered at the zero boundary is $\sum_{n=0}^{m} \binom{m}{n}$ and at the maximum queue length boundary the number is $\sum_{n=1}^{m} \binom{m}{n}$ where m is the number of centers. These increase rapidly, to say the least, and make any hand analysis cumbersome.

From these two models, however, several points can be noted. A general solution for the n-center model can be hypothezied. In general, we denote the attainment of maximum length by an "_" (an underbar), then

$$P_{a,b,c,\ldots,n} = \rho_1^{a}\rho_2^{b}\cdots\rho_T^{n}P_{0,\ldots,0} \text{ where } \rho_T = (\alpha_1+\alpha_2+\cdots+\alpha_i+\beta)/\mu \tag{42}$$

$$P_{\underline{a},b,\ldots,n} = \rho_1^{\underline{a}}\rho_2^{b}\cdots\rho_i^{m}\rho_j^{n}P_{0,\ldots,0} \text{ where } \rho_j=(\beta+\alpha_2+\cdots+\alpha_i)/\mu \tag{43}$$

$$P_{a,\underline{b},c,\ldots,n} = \rho_1^{a}\rho_2^{\underline{b}}\rho_3^{c}\cdots\rho_i^{m}\rho_j^{n}P_{0,\ldots 0} \text{ where } \rho_j=(\beta+\alpha_1+\alpha_3+\cdots+\alpha_i)/\mu \tag{44}$$

$$P_{\underline{a},\underline{b},c,\ldots,n} = \rho_1^{\underline{a}}\rho_2^{\underline{b}}\rho_3^{c}\cdots\rho_i^{m}\rho_j^{n}P_{0,\ldots,0} \text{ where } \rho_j=(\beta+\alpha_3+\cdots+\alpha_i)/\mu \tag{45}$$

$$P_{\underline{a},\underline{b},\ldots,\underline{i},n} = \rho_1^{\underline{a}}\rho_2^{\underline{b}}\cdots\rho_i^{\underline{m}}\rho_j^{n}P_{0,\ldots,0} \text{ where } \rho_j=\beta/\mu \tag{46}$$

From which the general trend in probabilities can be deduced. If one or more of the n streams has attained its maximum length that term in ρ_j is deleted. The only exception to this is the stream with input parameter β (the processing center), a maximum length does not delete β from the ρ_j expression.

For example, consider equation (43). What this implies is that if a center other than the processing center has attained its maximum length, then the input stream to the processing center for the duration of this maximum length has in effect n-1 streams in place of its original n streams. In general, for i maximum lengths, there are n-i streams in place of n streams to the processing center. Note this is only for maximum length boundary conditions at the sending centers.

Throughout the previous development, the existence of the probability matrix A has been ignored. This shall now be incorporated into the model. Figure 2.1 illustrates the queueing situation with the implementation of the probability matrix A for the three center model. This is a more accurate picture of what occurs within the network since not all jobs at center 1 and 3 will be transmitted to center 2. Some portion will be sent to the other centers. It is, therefore, a more realistic description of the function of the transmitter to add the probability matrix A to the model.

Each job which enters the transmitter's queue has a probability a_{ij} of being sent from center i to center j and a collective probability of $(1 - a_{ij})$ of being sent elsewhere. This added restriction does not affect the results mentioned earlier. The output of an exponential server is still Poisson with Poisson input and an infinite queue [7]. Therefore, adding the probability a_{ij} in effect splits the Poisson stream into two parts, both of which are still Poisson.

The general solution then becomes for the three center model:

$$P_{k,M,n} = \rho_1^{k} \rho_3^{M} \hat{\rho}_i^{n} P_{0,0,0} \qquad 0 \leq k < K \ , \ m=M \ , \ 0 \leq n \leq N$$

$$P_{K,m,n} = \rho_1^{K} \rho_3^{m} \hat{\rho}_j^{n} P_{0,0,0} \qquad k=k \ . \ 0 \leq m < M \ , \ 0 \leq n \leq N$$

$$P_{K,M,n} = \rho_1^{K} \rho_3^{M} \rho_2^{n} P_{0,0,0} \qquad k=k \ , \ m=M \ , \ 0 \leq n \leq N$$

$$P_{k,m,n} = \rho_1^{k} \rho_3^{m} \hat{\rho}_T^{n} P_{0,0,0} \qquad 0 \leq k < K \ , \ 0 \leq m < M \ , \ 0 \leq n \leq N$$

where the following definitions are used:

1. $\hat{\rho}_T$ = total traffic intensity for the processing center, center 2

$$\hat{\rho}_T = (\alpha_1 a_{12} + \alpha_3 a_{32} + \beta_2)/\mu_2 .$$

2. $\hat{\rho}_i$ = traffic intensity with center 3 blocked

$$\hat{\rho}_i = (\alpha_1 a_{12} + \beta_2)/\mu_2 .$$

3. $\hat{\rho}_j$ = traffic intensity with center 1 blocked

$$\hat{\rho}_j = (\alpha_3 a_{32} + \beta_2)/\mu_2 .$$

In other words, center 2 is no longer seeing all of the original stream from the transmitter, but only part of it. Therefore, with this change the previous results describe the model. In an analagous manner, these comments can be implemented in the n-center model to yield the following:

$$P_{a,b,\ldots,n} = \rho_1^{a}\rho_2^{b}\cdots\hat{\rho}_T^{\,n}P_{0,\ldots,0} \text{ where } \hat{\rho}_T = (\alpha_1 a_{1n} + \cdots + \alpha_i a_{in} + \beta)/\mu$$

$$P_{\underline{a},b,\ldots,n} = \rho_1^{\underline{a}}\rho_2^{b}\cdots\hat{\rho}_j^{\,n}P_{0,\ldots,0} \text{ where } \hat{\rho}_j = (\alpha_2 a_{2n} + \cdots + \alpha_i a_{in} + \beta)/\mu$$

$$P_{a,\underline{b},c,\ldots,n} = \rho_1^{a}\rho_2^{\underline{b}}\rho_3^{c}\cdots\rho_i^{m}\hat{\rho}_j^{\,n}P_{0,\ldots,0} \text{ where } \hat{\rho}_j = (\alpha_1 a_{1n} + \alpha_3 a_{3n} + \cdots + \alpha_i a_{in} + \beta)/\mu$$

$$P_{\underline{a},\underline{b},c,\ldots,n} = \rho_1^{\underline{a}}\rho_2^{\underline{b}}\rho_3^{c}\cdots\rho_i^{m}\hat{\rho}_j^{\,n}P_{0,\ldots,0} \text{ where } \hat{\rho}_j = (\alpha_3 a_{3n} + \cdots + \alpha_i a_{in} + \beta)/\mu$$

$$P_{\underline{a},\underline{b},\underline{c},\ldots,\underline{i},n} = \rho_1^{\underline{a}}\rho_2^{\underline{b}}\rho_3^{\underline{c}}\cdots\rho_i^{\underline{m}}\rho_j^{\,n}P_{0,\ldots,0} \text{ where } \rho_j = \beta/\mu$$

With these comments and for the reasons cited earlier, further development of the n-center model is left to future research.

4. PRIORITY QUEUEING

4.1. System Description

In the preceding analysis, the assumption has been that jobs are serviced in the order of arrival or first-come, first-served. This need not always be the case as there exist many situations where higher priority jobs may displace lower priority jobs in the queue. This is true, for instance, in freight handling where perishable goods have priority over non-perishable goods and machine maintenance where critical maintenance preempts any non-critical maintenance on the work schedules. The overall effect is that the waiting time for the lower priority jobs is increased as it experiences this displacement by the higher priority jobs.

This concept has been applied with little modification to a computer center. In every system, there will be some jobs demanding a faster turnaround than others, since by the very nature of the job, not every job requires the same turnaround. Using this knowledge can make each center in the network more responsive to those jobs requiring a faster turnaround at the expense of those jobs requiring a slower turnaround. One way of implementing such a scheme is through the use of priorities. Jobs requiring immediate turnaround can be given a higher priority than those jobs to which turnaround is not so essential. At the same time as the increase in priority a corresponding increase in computer charges can occur. In other words, the center manager may choose to implement a pay-for-priority scheme, with the obvious advantage of making more of the users happy more of the time.

To implement this scheme, we will take the two node model discussed in Chapter 2 and overlay a priority scheme on the arrivals to the transmitter and the processor. We shall assume that the priority assignment is universal throughout the network, but we shall restrict our attention to a particular center's view of these jobs. In other words, a priority 1 job has priority 1 in all centers, but in particular, our model is considering the effect of priority 1 jobs in the transmitter or processor separately. Figure 4.1 is an illustration of the priority model under consideration, note the similarity with Figure 2.3. Under the priority scheme, jobs of a lower priority may be displaced by jobs of a higher priority in the queue. Once any job has entered the server no preemption is permitted. Among jobs of equal priority in the queue, service is first-come, first-served.

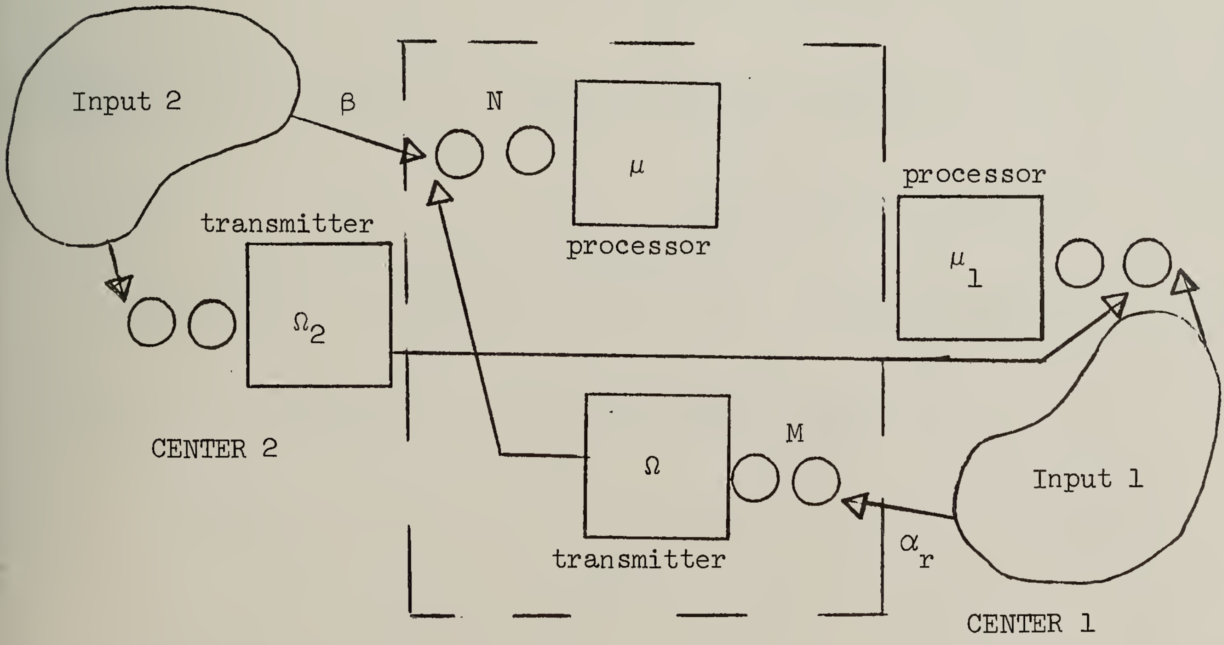


Figure 4.1. Two Center Network with Transmission Priorities

Let β be the arrival rate to the processor at center 2 and α_r the arrival rate to the transmitter at center 1 for jobs of priority r $(1 \leq r \leq R)$.

Let Ω be the service rate for jobs at the transmitter in center 1 and μ, the service rate for the processor at center 2. The total number of priority classes is given by R, and the queues for system 1 and 2 are bounded by $M > 1$ and $N > 1$, respectively. Then, to the definitions given earlier we add the following (where 1 is a high priority and R is a low priority assignment):

1. Total arrival rate to the transmitter of all priority classes.

$$\alpha = \sum_{r=1}^{R} \alpha_r \ .$$

2. Arrival rate for jobs of priority r or higher to the transmitter.

$$\lambda_r = \sum_{k=1}^{r} \alpha_k \ .$$

3. Traffic intensity for priority r or higher jobs in system 1.

$$\rho_1(r) = \lambda_r / \Omega \ .$$

4. Total traffic intensity for priority r or higher jobs in system 2.

$$\rho_T(r) = (\beta + \lambda_r)/\mu \ .$$

Let $P_{m,r,n}(t)$ denote the probability that there are exactly (m-1) units of priority r or higher in system 1 at time t and exactly n units in system 2 at time t. That is, the job arriving at time t to system 1 is $m^{\underline{th}}$ away from service. Note that the set $[(1-p_1(>0), P_{m,r,n}]$ totally characterizes the system at steady state.

The differential difference equations for the system are determined as before and at steady state we obtain the following:

$$0 = -(\lambda_r + \beta + \Omega)P_{m,r,0} + \lambda_r P_{m-1,r,0} + \mu P_{m,r,1} \qquad (47)$$

$$0 = -(\beta + \Omega + \mu + \lambda_r)P_{m,r,n} + \Omega P_{m+1,r,n-1} + \beta P_{m,r,n-1}$$
$$+ \mu P_{m,r,n+1} + \lambda_r P_{m-1,r,n} \qquad 0 \le m < M+1, 0 \le n < N \qquad (48)$$

$$0 = -(\Omega + \mu + \lambda_r)P_{m,r,N} + \lambda_r P_{m-1,r,N} + \Omega P_{m+1,r,N}$$
$$+ \beta P_{m,r,N-1} + \Omega P_{m+1,r,N} \qquad 0 \le m < M+1, n=N \qquad (49)$$

$$0 = -(\lambda_r + \beta)P_{1,r,0} + \mu P_{1,r,1} + \alpha P_{0,0} - \Omega P_{1,0} \qquad (50)$$

$$0 = -(\Omega + \lambda_r + \beta)P_{M+1,r,0} + \lambda_r P_{M,r,0} + \mu P_{M+1,r,1} \qquad m=M+1 \qquad (51)$$

$$0 = -(\Omega + \mu)P_{M+1,r,N} + \beta P_{M+1,r,N-1} + \lambda_r P_{M,r,N} \qquad m=M+1, n=N \qquad (52)$$

$$0 = -(\beta + \Omega + \mu)P_{M+1,r,n} + \mu P_{M+1,r,n+1} + \beta P_{M+1,r,n-1}$$
$$+ \lambda_r P_{M,r,n} \qquad m=M+1, 0 \le n < N \qquad (53)$$

Solving for the high order terms in equations (47) - (53) and putting the results in a closed form solution yields for system 1:

$$P_{m,r,n} = \rho_1(r)^{m-1}\ \rho_T(r)^n P_{1,r,0} \qquad 0\leq m<M+1,\ \ 0\leq n\leq N$$

$$= \rho_1(r)^M\ \rho_2{}^n P_{1,r,0} \qquad m=M+1,\ \ 0\leq n\leq N$$

Let $p_1(>0)$ denote the probability of not going directly into service for system 1, then

$$(1-p_1(>0)) + \sum_{m=1}^{M+1}\ \sum_{n=0}^{N}\ P_{m,r,n} = 1 \quad \text{and}$$

$$P_{1,r,0} = \left[\frac{(1-\rho_T(r)^{N+1})(1-\rho_1(r)^M)}{(1-\rho_T(r))\ (1-\rho_1(r))} + \frac{\rho_1(r)^M(1-\rho_2{}^{N+1})}{(1-\rho_2)}\right]^{-1} p_1(>0) \quad . \tag{54}$$

4.2. System Descriptors

Average number of priority r or higher in the queue for system 1,

$$Lq_1(r) = \sum_{m=0}^{M}\ \sum_{n=0}^{N}\ P_{m+1,r,n}$$

$$= \left[\frac{(1-\rho_T(r)^{N+1})[\rho_1(r)(1-\rho_1(r)^M) - M\rho_1(r)(1-\rho_1(r))]}{(1-\rho_T(r))(1-\rho_1(r))^2} + \frac{M\rho_1(r)^M(1-\rho_2{}^{N+1})}{(1-\rho_2)}\right] P_{1,r,0}$$

Average number of priority r in the queue for system 1,

$$Lqr_1 = Lq_1(r) - Lq_1(r-1) = \Bigg[(1-\rho_T(r)^{N+1})\,\frac{[\rho_1(r)(1-\rho_1(r)^M)-M\rho_1(r)^M(1-\rho_1(r))]}{(1-\rho_T(r))(1-\rho_1(r))^2}$$

$$-\,(1-\rho_T(r-1)^{N+1})\,\frac{[\rho_1(r-1)(1-\rho_1(r-1)^M)-M\rho_1(r-1)^M(1-\rho_1(r-1))]}{(1-\rho_T(r-1))(1-\rho_1(r-1))^2}$$

$$+\,\frac{M(\rho_1(r)^M-\rho_1(r-1)^M)(1-\rho_2^{N+1})}{(1-\rho_2)}\Bigg]\,P_{1,r,0} \qquad r\neq 1$$

Trivially, $Lql_1 = Lq_1(1)$.

For a priority r job, given that it is not displaced, the waiting times are derived by the following argument. When a job enters the queue, there are η_r units ahead of it. From this must be subtracted the term

$\sum_{n=0}^{N} (M+1)P_{M+1,r,n}$, to account for the situation that when the job arrives the queue can already be full of priority r or higher jobs.

$$\eta_r = (Lq_1(r) - \sum_{n=0}^{N} (M+1)P_{M+1,r,n})/p_1(>0)$$

The job must still wait for the job in service once it has waited for these η_r jobs in the queue. Therefore, the total waiting time, given that the job waits and is not displaced, is

$$W_r = (\eta_r + 1)/\Omega$$

Removing the condition on waiting,

$$W_r = [(Lq_1(r) - \sum_{n=0}^{N} (M+1)P_{M+1,r,n}) + p_1(>0)]/\Omega$$

$$= 1/\Omega \left[(1-\rho_T(r)^{N+1}) \frac{[\rho_1(r)(1-\rho_1(r)^M) - M\rho_1(r)^M(1-\rho_1(r))]}{(1-\rho_T(r))(1-\rho_1(r))^2} \right] P_{1,r,0} \quad (55)$$

For a priority 1 job, this is the waiting time for the tagged job since it cannot be displaced. For a priority $r \neq 1$ job, the waiting time becomes infinite, if it is displaced from the queue. If it is not displaced from the queue, then it must wait in addition for those higher priority jobs that arrive while it is in the queue and that take a position in front of the tagged job. Since the higher priority units arrive at a rate λ_{r-1} for the duration of his waiting time W_r, the job must wait an additional time of $\lambda_{r-1} W_r/\mu$. Therefore, denoting equation (55) by Z, the waiting time becomes

$$W_r = Z + \lambda_{r-1} W_r/\mu$$

$$= Z/(1-\rho_1(r-1)) .$$

4.3. The Processing System - System 2

For system 2, the processor, a similar argument can be used to derive the following solution. Let β_r be the arrival rate to the processor at center 2 for priority r jobs $(1 \leq r \leq R)$ and α, the arrival rate to the transmitter at center 1. Note earlier that β_i referred to the arrival for the $i\underline{th}$ center, the contex should clarify the meaning intended. Let Ω_r be the service rate for jobs of priority r at the transmitter in center 1, and μ the service rate for the processor at center 2. Units of priority r in system 1 exist even though priorities are not in use since the priority assignment is universal. The total number of priority classes is given by R, and the queues for system 1 and 2 are bounded by $M>1$ and $N>1$, respectively. The following definitions hold:

1. Total arrival rate to the processor of all priority classes,

$$\beta = \sum_{r=1}^{R} \beta_r .$$

2. Arrival rate to the processor for units of priority r or higher,

$$\sigma_r = \sum_{k=1}^{r} \beta_k , \text{ where } \beta = \sigma_R .$$

3. Traffic intensity for jobs of priority r or higher in system 2, exclusive of transmissions,

$$\rho_2(r) = \sigma_r/\mu .$$

4. Total traffic intensity for system 2 of priority r or higher jobs,

$$\rho_T(r) = (\alpha + \sigma_r)/\mu .$$

5. Total traffic intensity for system 1 for jobs of priority r or higher,

$$\rho_1(r) = \alpha/\Omega_r .$$

Let $P_{m,n,r}(t)$ denote the probability that at time t there are exactly m units in system 1 and exactly (n-1) units of priority r or higher in system 2, then

$$P_{m,n,r} = \rho_1(r)^m \rho_T(r)^{n-1} P_{0,1,r} \qquad 0 \leq m < M, \quad 0 \leq n \leq N+1$$

$$= \rho_1(r)^M \rho_2(r)^{n-1} P_{0,1,r} \qquad m=M, \quad 0 \leq n \leq N+1$$

Let $p_2(>0)$ be the probability of not going directly into service for system 2. Then using a similar argument as used to derive equation (54) we obtain:

$$P_{0,1,r} = \left[\frac{(1-\rho_1(r)^M)(1-\rho_T(r)^{N+1})}{(1-\rho_1(r))(1-\rho_T(r))} + \rho_1(r)^M \frac{(1-\rho_2(r)^{N+1})}{(1-\rho_2(r))} \right]^{-1} p_2(>0)$$

4.4. System Descriptors

Average number of priority r or higher in the queue for system 2,

$$Lq_2(r) = \sum_{m=0}^{M} \sum_{n=0}^{N} nP_{m,n+1,r}$$

$$= \left[\frac{(1-\rho_1(r)^M)[\rho_T(r) - (N+1)\rho_T(r)^{N+1} + N\rho_T(r)^{N+2}]}{(1-\rho_1(r))(1-\rho_T(r))} \right.$$

$$\left. + \rho_1(r)^M \frac{[\rho_2(r) - (N+1)\rho_2(r)^{N+1} + N\rho_2(r)^{N+2}]}{(1-\rho_2(r))} \right] P_{0,1,r}$$

Average number of priority r jobs in the queue for system 2, where trivially $Lq_2 1 = Lq_2(1)$.

$$Lq_2 r = Lq_2(r) - Lq_2(r-1) =$$

$$\left[(1-\rho_1(r)^M) \frac{[\rho_T(r)-(N+1)\rho_T(r)^{N+1}+N\rho_T(r)^{N+2}]}{(1-\rho_T(r))(1-\rho_1(r))} \right.$$

$$+ \frac{[\rho_T(r-1)-(N+1)\rho_T(r-1)^{N+1}+N\rho_T(r-1)^{N+2}]}{(1-\rho_1(r-1))(1-\rho_T(r-1))} (1-\rho_1(r-1)^M)$$

$$+ \rho_1(r)^M \frac{[\rho_2(r)-(N+1)\rho_2(r)^{N+1}+N\rho_2(r)^{N+2}]}{(1-\rho_2(r))}$$

$$\left. - \rho_1(r-1)^M \frac{[\rho_2(r-1)-(N+1)\rho_2(r-1)^{N+1}+N\rho_2(r-1)^{N+2}]}{(1-\rho_2(r-1))} \right] p_{0,1,r} \qquad r \neq 1$$

In an analogous manner to system 1, for a priority r job that is not displaced from the system, η_r is the number of jobs in the queue when the jobs enter the queue. Given that it waits,

$$\eta_r = (Lq_2(r) - (N+1) \sum_{m=0}^{M} P_{m,N+1,r})/p_2(>0),$$

where the term $(N+1)P_{m,N+1,\ r}$ accounts for the situation where a job cannot enter the queue since it is full. Then given that it waits,

$$\begin{aligned} W_r &= (\eta_r + 1)/\mu \\ &= (\eta_r + p_2(>0))/\mu \end{aligned}$$

$$= 1/\mu \left[(1-\rho_1(r)^M) \frac{[\rho_T(r)-(N+1)\rho_T(r)^N + N\rho_T(r)^{N+2}]}{(1-\rho_T(r))(1-\rho_1(r))} \right.$$

$$+ \rho_1(r)^M \frac{[\rho_2(r)-(N+1)\rho_2(r)^N + N\rho_2(r)^{N+2}]}{(1-\rho_2(r))}$$

$$\left. + \frac{(1-\rho_1(r)^M)(1-\rho_T(r)^{N+1})}{(1-\rho_1(r))(1-\rho_T(r))} + \frac{\rho_1(r)^M(1-\rho_2(r)^{N+1})}{(1-\rho_2(r))} \right] P_{0,1,r} \qquad (56)$$

For a priority 1 job, this is the total waiting time since it cannot be displaced once it enters the queue. For a priority r≠1 job, if it is displaced from the queue, its total waiting time is infinite. If it is not displaced from the queue, it must wait an additional amount $\sigma_{r-1} W_r/\Omega$, following the analysis given for system 1 waiting time statistics. The total waiting time for priority r≠1 job then becomes, denoting equation (56) by Y:

$$W_r = Y/(1-\rho_2(r-1)) \ .$$

5. NETWORK TOPOLOGY

In Chapter 1, three classes of networks were defined. In this chapter we will take a closer look at a representative of each of the topological classes: star, distributed, and ring structure. The discussion of each network is in chronological order by date of implementation.

From this survey, the parallels between functioning networks and the analysis presented earlier should become apparent. Moreover, the need and the direction for future analysis should also be clear.

5.1. The TUCC Network

1964 marked the start of the Triangle Universities Computation Center (TUCC) network. TUCC is a regional network consisting of an IBM 360/165 at Research Triangle Park (RTC), North Carolina, an IBM 360/75 at the University of North Carolina, Chapel Hill, and two IBM 360/40's at Duke University at Durham and North Carolina State University at Raleigh. In addition, 40 institutions are connected by terminals to the network. Each of the campus centers at Duke, North Carolina State and the University of North Carolina view TUCC as a pipeline through which they get massive computing power to service their users. Most of the communication is to the Research Triangle Park and there is little, if any, intracenter communication otherwise. Consequently, direct center to center links were not provided. Figure 5.1 illustrates the star topology characteristic of the TUCC network.

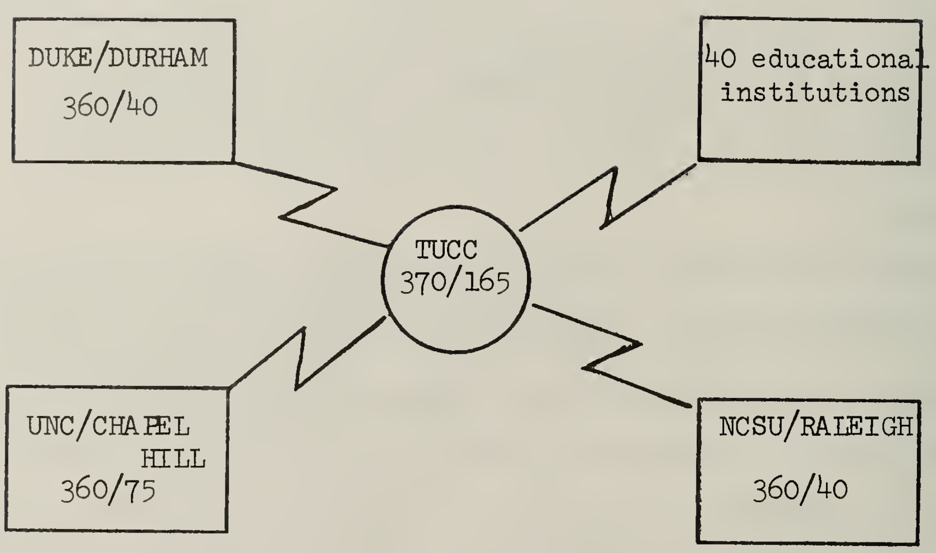


Figure 5.1. The TUCC Network

The primary motivation for the development of TUCC was economics: To give each of the institutions access to more computing power at lower cost than they could provide individually. One of the universities concluded that it would cost them about $19,000 per month more in hardware and personnel costs to provide all their computing services on campus than it would cost to continue participation in TUCC. This would represent a 40 percent increase over their present expense for terminal machine communications, and their share of TUCC expense.

There are other obvious advantages. First, application programs and ideas are easily exchanged and shared. A tangible savings is that single copies of computer software packages can be purchased and made available at one site for general use. In this way, duplication of programming and its cost is avoided since once a program is developed it can be used anywhere in the network without difficulty. Secondly, the member universities are better able to attract and keep faculty members who require large-scale computing facilities for research and teaching purposes. This is a tangible advantage not only for the large universities, but also for the smaller member institutions.

Individual machine characteristics are used to greater advantage and profit at considerable cost savings to all users, since a general need exists in many installations to access large or perhaps differently configured systems for parts of their regular batch computing needs. This, in turn, implies more efficient utilization of the total resources and improved turnaround through greater system balance.

A third advantage is the ability to attract competent system programmers and managers. In general, these personnel could not have been attracted to work in the environment of the individual institutions because of salary requirements and because of system sophistication considerations.

For the most part, communication is over low to medium speed terminals to the RTC. The OUTWATS line presently in use allows output to queue up for all active dial-up terminals. Once the job is received at

RTC its position in the processor queue is determined by a TUCC priority selection algorithm. The basic elements in the determination of a job's priority are:

1. amount of computer time the center has used for the day,
2. time submitted,
3. specified priority.

Jobs from one center compete with each other for queue position, while between centers high or low utilization at RTC determines the job's relative position in the queue. The founding universities have an additional bonus because the algorithm defaults unused resources to one or more of them dependent on demand. This is especially effective if their peak demands do not coincide. This flexibility is a major advantage to any network user. (For greater detail, see in particular [6, 29, 31, and 37 to 40].)

The relation between TUCC and the two-node network of Chapter 2 is obvious. Most communication in TUCC is terminal to processor or the reverse, where the important feature is transmission and processor service times. Since the major emphasis in TUCC has been economic, improved system performance and any benefits that might be reaped from it are of general interest.

5.2. The ARPA Network

The Advanced Research Projects Agency (ARPA) network began two years after TUCC. This is a large national computer network which views itself as a set of autonomous independent computer systems, interconnected so as to permit interactive resource sharing between any pair of systems. In other words, any program should be able to call on the resources of other computers much as it would a subroutine. Time-sharing systems with the associated software sharing are already prevalent within a local community. This means that for ARPA to be feasible it must offer such sharing on a nationwide basis in such a way that communication time is transparent to the users.

Before the ARPA topology was chosen (Figure 5.2) careful consideration was given to each of the other topological classes. The star configuration was enticing when the small response time was considered, but was at a disadvantage when reliability and alternate routes were demanded. The ring structure satisfied the reliability requirements, but failed to satisfy the two second response time stipulation. The result was the distributed network, combining the best characteristics of both. To protect against total line failure, there are at least two physically separate paths to route each message and round trip delay, because of the multiple connections, was within the two second limit. Thus, the topology of ARPA was selected to minimize the cost, maximize growth potential and yet satisfy all the design criteria.

The versatility of ARPA stems from the wide variation in the resources at each node. As many unique resources (which are only available at a small number of centers) as possible rather than common resources (which are available at many centers) were included in the initial twenty

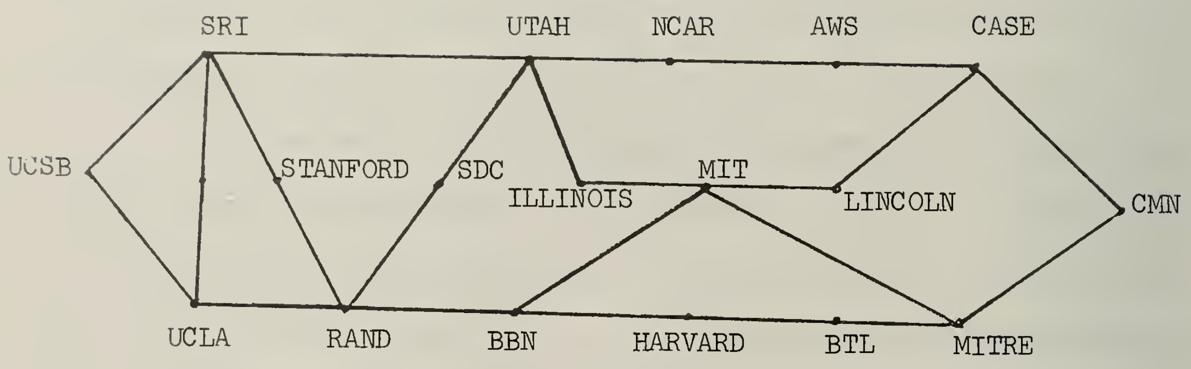


Figure 5.2. The ARPA Network

node configuration. In conjunction with the resources available, the nodes were chosen so that the communication cost was less than 25 percent of the computing costs of the system connected through the network.

Study of the communication technology showed that the use of conventional line switching facilities would be economically and technically inefficient. To resolve this difficulty ARPA decided to build a new kind of digital communications system employing wideband leased lines and message switching, where a path is not established in advance and each message carries an address. Since ARPA is not a fully connected network, messages normally traverse several nodes before reaching their destination. With this in mind, at each node a copy of the message is stored until it is safely received at the following node. The network is thus a store and forward system. However, with this decision come problems of routing, buffering, reliability and many more. To protect the centers from these network problems as well as to protect the network from center problems, each node has a small processor which is interconnected with the remaining network to form a subnet and to connect each center into the network. In this arrangement, the centers are called Hosts and the processors, Interface Message Processors (IMPs). This in effect divides the network into two parts, a subnet and the Hosts.

To complete the general discussion of ARPA, we will consider message delay in ARPA. The subnet transmits messages over unidirectional logical paths between Hosts, known as links. On a given link at any one time, only one message may be accepted and the link is blocked to all other traffic, until the message has been received and acknowledged. Since each Host has more than one link all traffic to or from that Host is not blocked.

Each IMP takes the message from its Host in segments, forms the segments into packets and ships each packet separately. At the destination the IMP reassembles the packets and delivers them to the Host as a single unit. The path of each message is determined by a routing algorithm which utilizes the total estimated transit time for its decisions. (This is further discussed in [8 and 21] and is only included in this discussion for completeness.)

Each IMP has a buffer storage space partitioned into about 70 fixed length buffers, each of which has the capability of storing a single packet. Message delay is subject to a variety to definitions. In ARPA delay measurements are based on time values at a single node by utilizing the notion of round trip times of messages and acknowledgments.

In a queueing model of ARPA conducted at U.C.L.A., a priority non-preemptive system of the model was investigated. Priorities were added mainly so that the effect of the higher priority acknowledgments and requests would be felt in contrast to ordinary message transmission. Service time was defined as the time required to transmit the message across the channel, and is dependent on the message length and the channel capacity. (For a more detailed description, see in particular [10].)

The intent of this section is to demonstrate the applicability of the theory developed in this paper. In ARPA the basis elements are those assumed to be present in any network: queues, transmission time delays (called service time in ARPA), and processing time at the center. This also demonstrates the strong correlation between reality and the theoretical model while reflecting the definite need for analysis. These analyses affect a better understanding of the advantages of networks and facilitate remedying network difficulties.

5.3. Distributed Computer Network

The third type of network discussed in Chapter 1 is the ring structure. This philosophy is exemplified in the Distributed Computer System (DCS). DCS is an experimental network system designed for the University of California at Irvine under the guidance of David J. Farber. In Farber's words, "the Distributed Computer System can be roughly sketched as a system of different processors interconnected by a common transmission system. Complete control of the system is distributed in time and space as the needs arise and as facilities are available."

In order to adequately understand DCS, it is necessary to discuss the overall aims of the project and the principles used to attain these aims. The design objectives proposed were reliability, low cost, incremental expansion capability, a variety of language systems, and modest system programming requirements. To meet these ends, a number of assumptions were made.

In order to provide a system that has a high probability of responding to a high percentage of users, implying that most users should receive what they request most of the time, the network may be designed such that it is vulnerable to partial failure. Nonetheless, the network should be designed so that it is invulnerable to total failure. That is, an effort should be made so that reliability is guaranteed for the network, while not necessarily for each component in the network.

To attain the goal of low cost, minicomputers were used on the nodes rather than larger processors such as IBM 360's. These smaller machines are flexible enough to provide the services required, while reducing the total network cost. Furthermore, in linking these smaller machines together, the system performance should improve as the work can now be done from a "pool" of resources. There is also a gain in that system performance no longer becomes dependent on whether or not a single machine is functioning but on whether the entire network is functioning.

Incremental expansion capability and a variety of language systems were specified since DCS involves a university community whose needs are continually changing. The network should be able to change with these needs. In this sense, a modular, regularized system has the added advantage that the configuration of computers or compilers can change and not affect the system performance as a whole.

The next question to be resolved is the actual configuration of the network and the principles used in its design. The DCS network attempts to solve this problem by allowing control of the network to reside in any processor which is free to do this task. Therefore, both the switching network and the control of the communications system is distributed. Furthermore, the network can be implemented at a reasonable cost since cables to each adjacent ring component need only be provided and not to the other n components. At worse, to maintain the reliability, cables to each adjacent and semi-adjacent node (node-skipping) need to be provided. This is still fewer connecting paths than required by a richer topology with a separate path between each component. Processors and languages are easily added around the ring and, therefore, expansion presents no additional problem. (The number of geographically distributed processors connected to the ring is expected to be approximately 30.) Finally, the ring topology is ascethically pleasing in that it is a simple structure.

The system provides for processors, file machines, and terminal controllers connected in the ring. Each of these components is connected through a ring interface to a unidirectional circular communications ring.

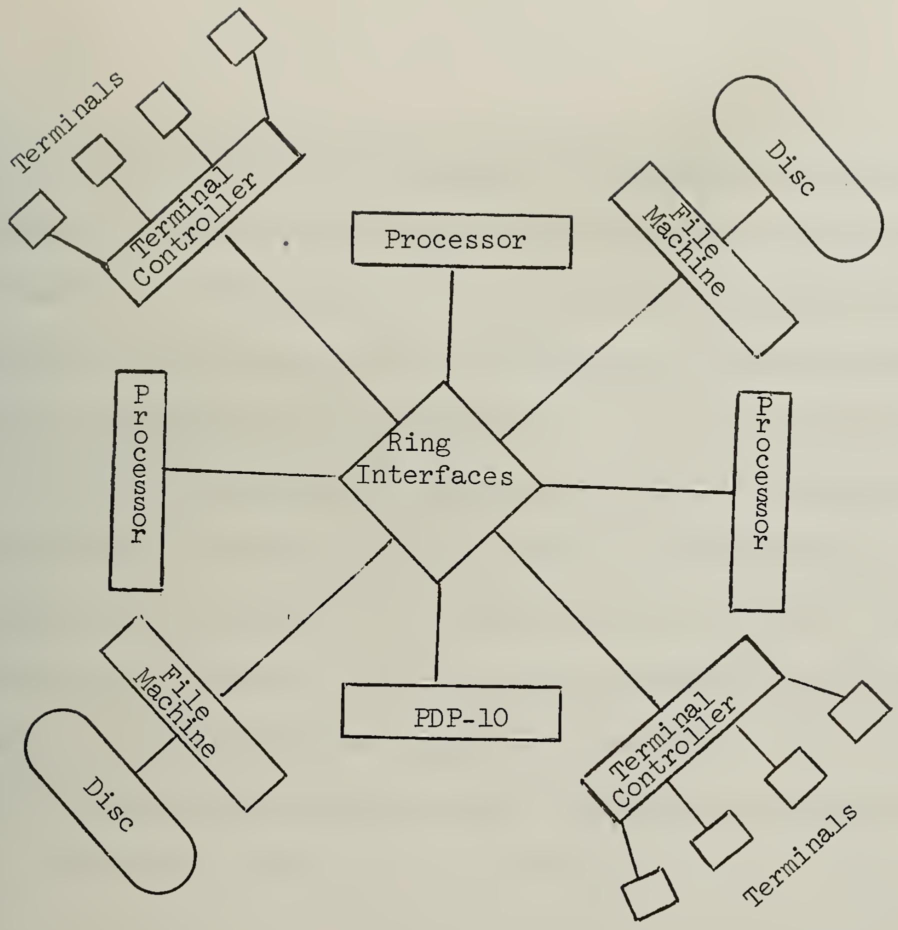


Figure 5.3. The Distributed Computer System

In addition, a PDP-10 is connected to the ring to facilitate testing of the system. From the above, it can be seen that the prototype system consists of minicomputers, ring interfaces, and interconnecting cable. There are seven minicomputers of approximately 16K capacity. Three of these share the processing load while two minicomputers with disks serve as file machines and two other minicomputers are terminal controllers for the connection of terminals to the system. All computing (except that performed in the file machines and by the terminal controllers) is performed by the three processing machines; thus if one fails, the other two can continue to provide service.

Messages are addressed to processes, not processors. This is accomplished through an associative store, the store contains the names of all processes and files active on the attached processor. When a message arrives over the ring, the destination process name is matched against the associative store. If a match occurs the message is copied and passed over the ring to the next node. The message is removed from the ring when it returns to the node where it originated. The ring itself may be thought of as a series of message slots. To transmit a message the node waits for an empty slot and when one is found, it places the message on the ring. The message is copied when necessary as it proceeds around the ring and checked against the original when it returns to the originating node. Note also that this scheme allows messages to be broadcast to all processes or a class of processes. Furthermore, it is extremely important to note that the process may physically be anywhere on the ring and may indeed have moved; the sender need not know this to correctly send a message.

All communication appears to be carried out along the ring as far as the processor is concerned since the Resident Message Router (RMR) routes messages from the ring onto the Incoming Message Queue (IMQ) and places messages to other processes into the Outgoing Message Queue (OMQ). Should a message on the OMQ be destined for a process active on the host, it is routed directly to the IMQ by the RMR and is not placed on the ring, although it appears as if it came from the ring. This scheme allows a process to keep the same address, its name, even though it has been moved from one host to another.

The parallels to the general network configuration discussed in Chapter 2 are strong. Each processor has an IMQ and an OMQ which corresponds to the transmission and processor queues at a single center. Transmission service time can be defined as the time it takes the message to travel around the ring including the time necessary for finding an idle slot. Processor service time is the time spent in service at the processor, exclusive of its time in the IMQ. With this clarification, the results derived earlier are applicable to this system, irrespective of the network topology. The essential elements, transmission and processors are present.

In this discussion, no mention was made of many aspects of the operating philosophy, such as garbage collection, idle slot detector, etc. The reader is referred to references [14] to [17] for a discussion of these aspects of DCS. Our purpose here is to outline the correlation between DCS and the general network configuration developed in this paper.

6. CONCLUSIONS

In this paper, we have taken the approach of looking at a network computer at double arm's length and seeing what the general network characteristics are. No consideration was given to any one center's or network's individual idiosyncrasies. To this end, we isolated the communication links and the computer center as the two distinguishing and vital elements in any network computer.

The philosophical approach in this paper has been that networks are now an everyday fact of life, and that there is a growing need to understand and improve on this fact. We have seen, in Chapter 2, the general model description and the degree of complexity involved in a two-center network. Chapter 3 provided a partial expansion of these results to the n-center network. Again the complexity and unwieldness of the equations increase rapidly. This highlights the need for some super analytical tool to give the analyst more insight into the intricacies of his work. Simulation is one such invaluable tool. The next step in this network center development would be to develop a simulation model to visually and graphically portray the relationships between the two center and the n-center networks. Moreover, alternate methods of queueing, such as priority and nonpriority queues, could be investigated with a minimum of effort. The work in any of these areas is by no stretch of the imagination complete. However, we do feel that a finger has been pointed down the correct road.

Network computers are economically feasible only if they can favorably compete as an alternative to standard computing techniques. To do so, system measures and guidelines have to be established. Chapter 2 through 4 represent alternative methods of network analysis. Chapter 5 was included to demonstrate the direct correlation between today's existing networks and the theoretical network discussed in this paper. For none of these networks, real or imaginary, is the development or analysis concluded.

APPENDIX A

The System (M, ∞)

Solution: $$P_{m,n} = \rho_1^{\,m}\rho_T^{\,n}P_{0,0} \qquad 0 \leq m < M \qquad \forall n$$

$$P_{M,n} = \rho_1^{\,M}\rho_2^{\,n}P_{0,0} \qquad m{=}M \qquad \forall n$$

1. $$P_{0,0}^{\;-1} = \frac{(1-\rho_2)(1-\rho_1^{\,M}) + \rho_1^{\,M}(1-\rho_1)(1-\rho_T)}{(1-\rho_T)(1-\rho_1)(1-\rho_2)}$$

System Descriptors:

2. $$L_s = \frac{P_{0,0}[(1-\rho_2)(\rho_T+\rho_1) + \rho_1^{\,M}(2(\rho_2-\rho_T) - \rho_2\rho_1(1-\rho_T)) - \rho_1^{\,M+1}(1-\rho_T)]}{(1-\rho_1)(1-\rho_2)(1-\rho_T)}$$

3. $$E(m) = \left[\frac{[\rho_1(1-\rho_1^{\,M}) - M\rho_1^{\,M}(1-\rho_1)]}{(1-\rho_1)^2(1-\rho_T)} + \frac{M\rho_1^{\,M}}{(1-\rho_2)}\right]P_{0,0}$$

4. $$E(n) = \left[\frac{(1-\rho_1^{\,M})\rho_T}{(1-\rho_1)(1-\rho_T)^2} + \frac{M\rho_1^{\,M}\rho_2}{(1-\rho_2)^2}\right]P_{0,0}$$

5. $$E(m,n) = P_{0,0}\left[(1-\rho_1)^2(1-\rho_T)^2\rho_1^{\,M}(1-(M-1)\rho_2) + (1-\rho_2)^2[(1-\rho_1^{\,M})(\rho_1(1-\rho_T) + \rho_T(1-\rho_1)) - (1-\rho_1)M\rho_1^{\,M}(1-\rho_T)]\right] \Big/ \left[(1-\rho_1)^2(1-\rho_T)^2(1-\rho_2^{\,2})\right]$$

6. $$L_{q1} = \left[\frac{\rho_1 2(1-\rho_1^{M-1}) - M\rho_1^{M}(1-\rho_1) + \rho_1^{M}(1-\rho_1)}{(1-\rho_T)(1-\rho_1)^2} + \frac{(M-1)\rho_1^{M}}{(1-\rho_2)}\right] P_{0,0}$$

7. $$L_{q2} = \left[\frac{(1-\rho_1^{M})\rho_T^{2}}{(1-\rho_1)(1-\rho_T)^2} + \frac{\rho_1^{M}\rho_2^{2}}{(1-\rho_2)^2}\right] P_{0,0}$$

8. $$p_1(>0) = P_{0,0}\left[\frac{\rho_1(1-\rho_1^{M-1})}{(1-\rho_T)(1-\rho_1)} + \frac{\rho_1 M}{(1-\rho_2)}\right]$$

9. $$p_2(>0) = P_{0,0}\left[\frac{(1-\rho_1^{M})\rho_T}{(1-\rho_1)(1-\rho_T)} + \frac{\rho_1^{M}\rho_2}{(1-\rho_2)}\right]$$

10. $$W_1 = P_{0,0}/\Omega\left[\frac{M\rho_1^{M}}{(1-\rho_2)} + \frac{\rho_1^{M}(1-\rho_1) - M\rho_1^{M}(1-\rho_1) + \rho_1(1-\rho_1^{M-1})}{(1-\rho_T)(1-\rho_1)^2}\right]$$

11. $$W_2 = P_{0,0}/\mu\left[\frac{\rho_1^{M}\rho_2}{(1-\rho_2)^2} + \frac{(1-\rho_1^{M})\rho_T}{(1-\rho_1)(1-\rho_T)^2}\right]$$

APPENDIX B

The System (∞, N)

Solution: $P_{m,n} = \rho_1^{\,m} \rho_T^{\,n} P_{0,0} \quad \forall m \quad 0 \leq n \leq N$

1. $P_{0,0}^{\,-1} = (1-\rho_T^{\,N+1})/(1-\rho_1)(1-\rho_T)$

System Descriptors:

2. $L_s = [\rho_T(1-\rho_T^{\,N}) + \rho_1(1-\rho_T^{\,N+1})]/(1-\rho_T^{\,N+1})$

3. $E(m) = \rho_1/(1-\rho_1)$

4. $$E(n) = \frac{[\rho_T(1-\rho_T^{\,N+1}) - (N+1)(1-\rho_T)\rho_T^{\,N+1}]}{(1-\rho_T)(1-\rho_T^{\,N+1})}$$

5. $$E(m,n) = \frac{\rho_1}{(1-\rho_1)} + \frac{\rho_T}{(1-\rho_T)} - \frac{N\rho_T^{\,N+1}}{(1-\rho_T^{\,N+1})}$$

6. $L_{q1} = \rho_1^{\,2}/(1-\rho_1)$

7. $$L_{q2} = \frac{\rho_T^{\,2}(1-\rho_T^{\,N}) - (N+1)\rho_T^{\,N+1}(1-\rho_T) + \rho_T^{\,N+1}(1-\rho_T)}{(1-\rho_T)(1-\rho_T^{\,N+1})}$$

8. $p_1(>0) = \rho_1$

9. $p_2(>0) = \rho_T(1-\rho_T^{\,N})/(1-\rho_T^{\,N+1})$

10. $W_1 = \rho_1/(\Omega - \alpha)$

11. $$W_2 = 1/\mu \, \frac{[\rho_T(1-\rho_T^{\,N}) - (N+1)\rho_T^{\,N+1}(1-\rho_T) + \rho_T^{\,N+1}(1-\rho_T)]}{(1-\rho_T)(1-\rho_T^{\,N+1})}$$

APPENDIX C

The System (∞, ∞)

Solution: $P_{m,n} = \rho_1{}^m \rho_T{}^N P_{0,0} \qquad \forall m \ \forall n$

1. $P_{0,0} = (1-\rho_1)(1-\rho_T)$

System Descriptors:

2. $L_s = \rho_1 + \rho_T$

3. $E(m) = \rho_1/(1-\rho_1)$

4. $E(n) = \rho_T/(1-\rho_T)$

5. $E(m,n) = \rho_1/(1-\rho_1) + \rho_T/(1-\rho_T)$

6. $L_{q1} = \rho_1{}^2/(1-\rho_1)$

7. $L_{q2} = \rho_T{}^2/(1-\rho_T)$

8. $p_1(>0) = \rho_1$

9. $p_2(>0) = \rho_T$

10. $W_1 = \rho_1/(\Omega-\alpha)$

11. $W_2 = \rho_T/(\mu-w)$

LIST OF REFERENCES

[1] Bowdon, E. K., Sr., "Modeling and Analysis of a Network of Computers," Ph.D. Dissertation, University of Iowa, Iowa City, Iowa, 1969.

[2] Bowdon, E. K., Sr., "Network Computer Analysis," Proc. of the Fifth International Conference on System Sciences, January 1972, pp. 201-203.

[3] Bowdon, E. K., Sr., "Priority Assignment in a Network of Computers," IEEE Transactions on Electronic Computers, November 1969.

[4] Bowdon, E. K., Sr., and Barr, W. J., "Cost Effective Priority Assignment in Network Computers," University of Illinois Department of Computer Science Report No. UIUCDCS-R-72-509, March 1972, 23 pages.

[5] Bowdon, E. K., Sr., and Barr, W. J., "Throughput Optimization in Network Computers," Proc. of the Fifth Hawaii International Conference on System Sciences, January 1972, pp. 528-530.

[6] Brooks, F. P., Ferrell, J. K., and Gallie, T. M., "Organizational Financial and Political Aspects of a Three University Computing Center," Proc. of IFIPS Congress, 1968, pp. 49-52.

[7] Burke, P. J., "The Output of a Queueing System," Operations Research, Vol. 4, 1956, pp. 699-704.

[8] Carr, C. S., Crocker, S. D., and Cerf, V. G., "Host-Host Communication Protocol in the ARPA Network," Proc. of the Spring Joint Comp. Conf., Vol. 36, 1970, pp. 589-597.

[9] Cobham, A., "Priority Assignments in Waiting Line Problems," Operations Research, Vol. 2, 1954, pp. 70-76, "A Correction," Operations Research, Vol. 3, 1955, p. 547.

[10] Cole, G. P., "Computer Network Measurements: Techniques and Experiments," University of California, Los Angeles School of Engineering and Applied Science. Report No. UCLA-ENG-7165, October 1971, 350 pages.

[11] Conway, R. W., Maxwell, W. L., and Miller, L. W., Theory of Scheduling, Addison-Wesley Publishing Company, Reading, Mass., 1967.

[12] Cox, D. R., and Smith, W. L., Queues, Methuen and Company, London, 1961.

[13] Dressin, S. A., and Reich, E., "Priority Assignment on a Waiting Line," Quarterly of Applied Mathematics, Vol. 15, 1957, pp. 208-211.

[14] Farber, D. J., "Networks: An Introduction," Datamation, Vol. 18, No. 4, April 1972, pp. 36-39.

[15] Farber, D. J., "Progress Report on the Distributed Computing System," University of California at Irvine, Technical Report, January 1972.

[16] Farber, D. J., and Larson, K. C., Supplement to Proposal for Research Submitted to the National Science Foundation on Distributed Computing System, University of California, Irvine, Technical Report, October 1970.

[17] Farber, D. J., and Larson, K. C., "The System Architecture of the Distributed Computer System--An Informal Description," University of California, Irvine, Technical Report No. 11, September 1971, 36 pages.

[18] Frank, H., and Frisch, I. T., Communication, Transmission and Transportation Networks, Addison-Wesley Publishing Company, Reading, Mass., 1971.

[19] Frank, H. et. al., "Topological Considerations in the Design of the ARPA Net," Proc. of the Spring Joint Comp. Conf., Vol. 36, 1970, pp. 581-587.

[20] Hadley, G., Linear Programming, Addison-Wesley, Reading, Mass., 1962.

[21] Heart, F. E., et. al., "The Interface Message Processor for the ARPA Computer Network," Proc. of the Spring Joint Comp. Conf., Vol. 36, 1970, pp. 551-567.

[22] Holley, J., "Waiting Line Subject to Priorities," Operations Research, Vol. 2, 1954, pp. 341-343.

[23] Hunt, G. C., "Sequential Arrays of Waiting Lines," Operations Research, Vol. 2, 1954, pp. 674-683.

[24] Jackson, J. R., "Networks of Waiting Lines," Operations Research, Vol. 5, 1957, pp. 518-521.

[25] Kleinrock, L., "Analytic and Simulation Methods in Computer Network Design," Proc. of the Spring Joint Comp. Conf., Vol. 36, 1970, pp. 569-579.

[26] Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay, McGraw-Hill, New York, 1964.

[27] Miller, R. G., Jr., "Priority Queues," Annals of Mathematical Statistics, Vol. 31, 1960, pp. 86-103.

[28] Morse, P. M., Queues, Inventories, and Maintenance, John Wiley and Sons, New York, 1958.

[29] Parker, L. T., Jr., Gallie, T. M., Brooks, F. P., Jr., and Ferrell, J. K., "Introducing Computing to Smaller Colleges and Universities--a Progress Report," Comm. ACM, Vol. 12, No. 6, June 1969, pp. 319-323.

[30] Prabhu, N. U., Queues and Inventories, Wiley and Sons, Inc., New York, 1965.

[31] Ragland, J. R., and Cilley, N. Y., "Report for Fiscal Year 1970-1971," Triangle Universities Computation Center, Research Triangle Park, October 1971, 33 pages.

[32] Reich, E., "Notes on Queues in Tandem," Annuals of Math. Statistics, Vol. 34, 1963, pp. 338-341.

[33] Reich, E., "Waiting Times When Queues are in Tandem," Annuals of Math. Statistics, Vol. 28, 1957, pp. 768-773.

[34] Roberts, L. G. and Wessler, B. D., "Computer Network Development to Achieve Resource Sharing," Proc. of Spring Joint Comp. Conf., Vol. 36, 1970, pp. 543-549.

[35] Saaty, T. L., Elements of Queueing Theory, McGraw-Hill, New York, 1961.

[36] Saaty, T. L., "Resume of Useful Formulas in Queueing Theory," Operations Research, Vol. 5, 1957, pp. 161-200.

[37] Stephenson, J. W., and Williams, L. H., "Hasp-to-Hasp Inter-Computer Communication Network of TUCC," Proc. of IEEE International Computer Society Conference, 1971, pp. 99-100.

[38] Walker, J. F., "TUCC Priority Selection Algorithm," TUCC Memorandum, No. IM-002-0, February 1971, 7 pages

[39] Williams, L. H., "Computers in Communications," Virginia Computer User's Conference, 1972, 5 pages.

[40] Williams, L. H., "A Functioning Computer Network for Higher Education in North Carolina," Proc. of the Fall Joint Computer Conference, Vol. 41, 1972, pp. 899-964.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-73-577	2.	3. Recipient's Accession No.

4. Title and Subtitle

Queues and Network Computers

5. Report Date
May 1973

6.

7. Author(s)
Linda Anne Da Rin Mills

8. Performing Organization Rept. No. UIUCDCS-R-73-577

9. Performing Organization Name and Address
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.
NSF GJ 28289

12. Sponsoring Organization Name and Address
National Science Foundation
1800 G Street, N.W.
Washington, D.C. 20550

13. Type of Report & Period Covered
Master of Science Thesis

14.

15. Supplementary Notes

16. Abstracts

With the advent of the network computer has come an increasing movement towards developing the necessary analytical tools to properly evaluate and improve these networks. One such tool is system modeling to assist in performance evaluation of networks and computer centers whose usefullness and, consequently, communication flow is daily increasing. While much has been said of performance evaluation for particular centers, little has been done concerning the analysis of the entire network. Our purpose is to postulate a queueing theory model of a theoretical network and to use this model to predict such measures as the number in the system, the number in the queue, and system efficiency.

7. Key Words and Document Analysis. 17a. Descriptors

Network Computer, Dual Server, Network Center, Star Network, Distributed Network, Ring Structure, Stochastic Process, Queueing Theory, Priority Queue.

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

. Availability Statement
opies may be requested from the address given in
9) above.

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page
UNCLASSIFIED

21. No. of Pages
74

22. Price
NA